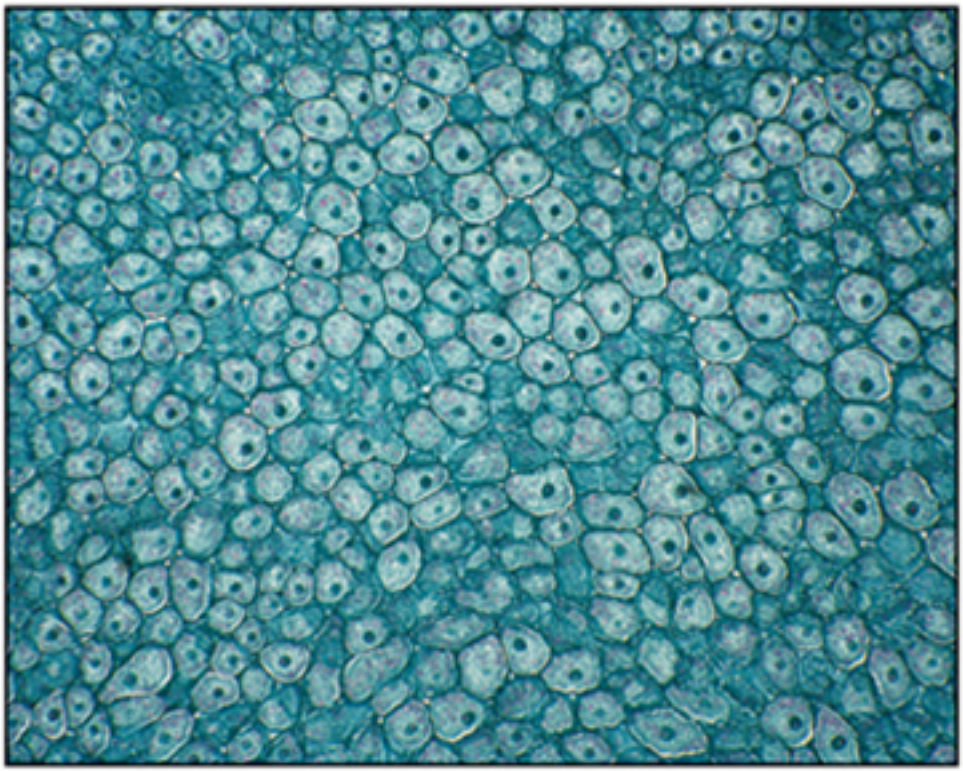
HANDBOOK OF RESEARCH ON

COMPUTATIONAL METHODOLOGIES IN GENE REGULATORY NETWORKS



Sanjoy Das, Doina Caragea, Stephen M. Welch & William H. Hsu

Handbook of Research on Computational Methodologies in Gene Regulatory Networks

Sanjoy Das
Kansas State University, USA

Doina Caragea
Kansas State University, USA

Stephen M. Welch
Michigan State University, USA

William H. Hsu
Kansas State University, USA

Medical Information Science Reference

MEDICAL INFORMATION SCIENCE REFERENCE

Hershey · New York

Director of Editorial Content: Kristin Klinger
Senior Managing Editor: Jamie Snavely
Assistant Managing Editor: Michael Brehm
Publishing Assistant: Sean Woznicki
Typesetter: Michael Brehm, Kurt Smith
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Medical Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

Library of Congress Cataloging-in-Publication Data

Handbook of research on computational methodologies in gene regulatory networks / Sanjoy Das ... [et al.], editors.
p. cm.
Includes bibliographical references and index.
Summary: "This book focuses on methods widely used in modeling gene networks including structure discovery, learning, and optimization"--Provided by publisher.
ISBN 978-1-60566-685-3 (hardcover) -- ISBN 978-1-60566-686-0 (ebook) 1. Genetic regulation--Mathematical models--Handbooks, manuals, etc. I. Das, Sanjoy, 1968-
QH450.H36 2010
572.8'65--dc22
2009017383

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

List of Reviewers

Manuel Barrio, *University of Valladolid, Spain*
Sebastian Bauer, *Charité Universitätsmedizin Berlin, Germany*
Daniel Bryce, *Utah State University, USA*
Kevin Burrage, *University of Queensland, Australia*
Doina Caragea, *Kansas State University, USA*
Adriana Climescu-Haulica, *Université Joseph Fourier, France*
Yang Dai, *University of Illinois at Chicago*
David Danks, *Carnegie Mellon University, USA*
Christian Darabos, *Université de Lausanne, Switzerland*
Alberto de la Fuente, *CRS4 Bioinformatica, Italy*
Chris Glymour, *Carnegie Mellon University, USA*
Angela Gon¸calves, *Darwin College, UK*
Mika Gustafsson, *Linköpings universitet, Sweden*
Ina Hoeschele, *Virginia Polytechnic Institute and State University, USA*
Jack Horner, *USA*
William H. Hsu, *Kansas State University, USA*
Lars Kaderali, *University of Heidelberg, Germany*
Ivan V. Ivanov, *Texas A&M University, USA*
Seungchan Kim, *Arizona State University, USA*
Ina Koch, *Max Planck Institute for Molecular Genetics, Germany*
Hiroyuki Kuwahara, *University of Trento Centre for Computational and Systems Biology, Italy*
Larry Liebovitch, *Florida Atlantic University, USA*
Bing Liu, *Monsanto Co., USA*
Michael Margaliot, *Tel Aviv University, Israel*
Yosihiro Mori, *Kyoto Institute of Technology, Japan*
Chris J. Myers, *University of Utah, USA*
Masahiro Okamoto, *Kyushu University, Japan*
Arlindo L. Oliveira, *Cadence Research Laboratories, USA*
Nicole Radde, *University of Leipzig, Germany*
Ramesh Ram, *Monash University, Australia*
Andre S. Ribeiro, *Tampere University of Technology, Finland*
David Sankoff, University of Ottawa, Canada
Till Steiner, *Honda Research Institute Europe GmbH, Germany*
Ala Ugo, *University of Turin, Turin, Italy*
Yong Wang, *Chinese Academy of Sciences, China*
Stephen M. Welch, *Kansas State University, USA*

List of Contributors

Table of Contents

Section 4
Structure and Parameter Learning

Section 5
Analysis & Complexity

Section 7
Network Simulation Studies

Section 8
Other Studies

Detailed Table of Contents

determine a network of interactions. In the first part of the chapter, we present a functional introduction to mechanisms involved in genes expression regulation, an overview of network theory, and main technologies developed in last years to analyze biological processes are discussed. In the second part, we review genes regulatory networks and their importance in system biology.

Section 2
Network Inference

Bayesian networks have become a commonly used tool for inferring structure of gene regulatory networks from gene expression data. In this framework, genes are mapped to nodes of a graph, and Bayesian techniques are used to determine a set of edges that best explain the data, that is to infer the underlying structure of the network. This chapter begins with an explanation of the mathematical framework of Bayesian networks in the context of reverse engineering of genetic networks. The second part of this review discusses a number of variations upon the basic methodology, including analysis of discrete vs. continuous data or static vs. dynamic Bayesian networks, different methods of exploring the potentially huge search space of network structures, and the use of priors to improve the prediction performance. This review concludes with a discussion of methods for evaluating the performance of network structure inference algorithms.

In this chapter, we address techniques that can be applied to establish causality between the various nodes in a GRN. These techniques are based on the joint analysis of DNA marker and expression as well as DNA sequence information. In addition to Bayesian networks, another modeling approach, statistical equation modeling, is discussed.

This chapter describes the model of genetic regulatory interactions. The model has a Boolean logic semantics representing the cooperative influence of regulators (activators and inhibitors) on the expression of a gene. The model is a probabilistic one, hence allowing for the statistical learning to infer the genetic interactions from microarray gene expression data. Bayesian approach to model inference is employed

enabling flexible definitions of a priori probability distributions of the model parameters. Markov Chain Monte Carlo (MCMC) simulation technique Gibbs sampling is used to facilitate Bayesian inference. The problem of identifying actual regulators of a gene from a high number of potential regulators is considered as a Bayesian variable selection task. Strategies for the definition of parameters reducing the parameter space and efficient MCMC sampling methods are the matter of the current research.

Differential equation models provide a detailed, quantitative description of transcription regulatory networks. However, due to the large number of model parameters, they are usually applicable to small networks only, with at most a few dozen genes. Moreover, they are not well suited to deal with noisy data. In this chapter, we show how to circumvent these limitations by integrating an ordinary differential equation model into a stochastic framework. The resulting model is then embedded into a Bayesian learning approach. We integrate the-biologically motivated-expectation of sparse connectivity in the network into the inference process using a specifically defined prior distribution on model parameters. The approach is evaluated on simulated data and a dataset of the transcriptional network governing the yeast cell cycle.

Section 3
Modeling Methods

As noise and delays are intrinsic to biochemical processes, they must be accounted for when dealing with the most detailed differential equation models of GRNs. The issue is addressed in this chapter. A basic Monte Carlo simulation technique to simulate noisy biochemical reactions, as well as a generalization to include delays, is described in this chapter. The chapter follows this with a study into 'coarse grain' approaches, which reduce computational costs when dealing with larger biochemical systems. The methodology is demonstrated with a few case studies.

phase of modelling. An illustrative example of the causal model for GRN modelling is also included and applied to model the yeast cell cycle dataset. The results obtained are compared for providing biological relevance to the findings which thereby underpins the CI based modelling techniques.

Section 4
Structure and Parameter Learning

Investigating gene regulatory networks is important to understand mechanisms of cellular functions. Recently, the synthesis of gene regulatory networks having desired functions has become of interest to many researchers because it is a complementary approach to understanding gene regulatory networks, and it could be the first step in controlling living cells. In this chapter, we discuss a synthesis problem in gene regulatory networks by network learning. The problem is to determine parameters of a gene regulatory network such that it possesses given gene expression pattern sequences as desired properties. We also discuss a controller synthesis method of gene regulatory networks. Some experiments illustrate the performance of this method.

The reconstruction of genetic regulatory networks from microarray gene expression measurements has been a challenging problem in bioinformatics. Various methods have been proposed for this problem including the Bayesian Network (BN) approach. In this chapter we provide a comprehensive survey of the current development of using structure priors derived from high-throughput experimental results such as protein-protein interactions, transcription factor binding location data, evolutionary relationships and literature database in learning regulatory networks.

Machine learning methods to find graphical models of genetic regulatory networks from cDNA microarray data have become increasingly popular in recent years. We provide three reasons to question the reliability of such methods: (1) a major theoretical challenge to any method using conditional independence relations; (2) a simulation study using realistic data that confirms the importance of the theoretical challenge; and (3) an analysis of the computational complexity of algorithms that avoid this theoretical challenge. We have no proof that one cannot possibly learn the structure of a genetic regulatory network from microarray data alone, nor do we think that such a proof is likely. However, the combination of (i) fundamental challenges from theory, (ii) practical evidence that those challenges arise in realistic data, and (iii) the difficulty of avoiding those challenges leads us to conclude that it is unlikely that current microarray technology will ever be successfully applied to this structure learning problem.

Section 5
Analysis & Complexity

Constructing computational models of genomic regulation faces several major challenges. While the advances in technology can help in obtaining more and better quality gene expression data, the complexity of the models that can be inferred from data is often high. This high complexity impedes the practical applications of such models, especially when one is interested in developing intervention strategies for disease control, for example, preventing tumor cells from entering a proliferative state. Thus, estimating the complexity of a model and designing strategies for complexity reduction become crucial in problems such as model selection, construction of tractable subnetwork models, and control of the dynamical behavior of the model. In this chapter, we discuss these issues in the setting of Boolean networks and probabilistic Boolean networks–two important classes of network models for genomic regulatory networks.

With advances in high throughput methods of data collection for gene regulatory networks, we are now in a position to face the challenge of elucidating how these genes coupled with environmental stimuli orchestrate the regulation of cell-level behaviors. Understanding the behavior of such complex systems is likely impossible to achieve with wet-lab experiments alone due to the amount and complexity of the data being collected. Therefore, it is essential to integrate the experimental work with efficient and accurate computational methods for analysis. Unfortunately, such analysis is complicated not only by the sheer size of the models of interest but also by the fact that gene regulatory networks often involve small

molecular counts making discrete and stochastic analysis necessary. To address this problem, this chapter presents a model abstraction methodology which systematically performs various model abstractions to reduce the complexity of computational biochemical models resulting in substantial improvements in analysis time with limited loss in accuracy.

A better understanding of the behavior of a cell, as a system, depends on our ability to model and understand the complex regulatory mechanisms that control gene expression. High level, qualitative models, of gene regulatory networks can be used to analyze and characterize the behavior of complex systems, and to provide important insights on the behavior of these systems. In this chapter, we describe a number of additional functionalities that, when supported by a symbolic model checker, make it possible to answer important questions about the nature of the state spaces of gene regulatory networks, such as the nature and size of attractors, and the characteristics of the basins of attraction. We illustrate the type of analysis that can be performed by applying an improved model checker to two well studied gene regulatory models, the network that controls the cell cycle in the yeast *S. cerevisiae*, and the network that regulates formation of the Dorsal-Ventral boundary in *D. melanogaster*. The results show that the insights provided by the analysis can be used to understand and improve the models, and to formulate hypotheses that are biologically relevant and that can be confirmed experimentally.

The expression of genes depends on the physical structure of DNA, how the function of DNA is regulated by the transcription factors expressed by other genes, RNA regulation such as that through RNA interference, and protein signals mediated by protein-protein interaction networks. We illustrate different approaches to determining information about the network of gene regulation from experimental data. First, we show that we can use statistical information of the mRNA expression values to determine the global topological properties of the gene regulatory network. Second, we show that analyzing the changes in expression due to mutations or different environmental conditions can give us information on the relative importance of the different mechanisms involved in gene regulation.

distribution present in microarrays. The main effort of this chapter is the exploration of a fully data driven approach to the integration problem in a "soft evidence" based way. Integration is here seen as the process of incorporation of uncertain a priori knowledge and is therefore only relied upon if it lowers the prediction error. An efficient implementation is carried out by a Linear Programming formulation. This LP problem is solved repeatedly with small modifications, from which we can benefit by restarting the primal simplex method from nearby solutions, which enables a computational efficient execution. We perform a case study for data from the yeast cell cycle, where all verified genes are putative regulators and the a priori knowledge consists of several types of binding data, text-mining, and annotation knowledge.

Section 7
Network Simulation Studies

Chapter 21

In this chapter, we describe the use of evolutionary methods for the in silico generation of artificial gene regulatory networks (GRNs). These usually serve as models for biological networks and can be used for enhancing analysis methods in biology. We clarify our motivation in adopting this strategy by showing the importance of detailed knowledge of all processes, especially the regulatory dynamics of interactions undertaken during gene expression. To illustrate how such a methodology works, two different approaches to the evolution of small-scale GRNs with specified functions, are briefly reviewed and discussed. Thereafter, we present an approach to evolve medium sized GRNs with the ability to produce stable multicellular growth. The computational method employed allows for a detailed analysis of the dynamics of the GRNs as well as their evolution. We have observed the emergence of negative feedback during the evolutionary process, and we suggest its implication to the mutational robustness of the regulatory network which is further supported by evidence observed in additional experiments.

Chapter 22

In this chapter, we propose a new model for Gene Regulatory Networks (GRN). The model incorporates more biological detail than other approaches, and is based on an artificial genome from which several products like genes, mRNA, miRNA, noncoding RNA, and proteins are extracted and connected, giving rise to a heterogeneous directed graph. We study the dynamics of the networks thus obtained, along with their topology (using degree distributions). Some considerations are made about the biological meaning of the outcome of the simulations.

Section 8
Other Studies

In this chapter, a computational formalism for modeling and reasoning about the control of biological processes is explored. It comprises five main sections: a survey of related work, a background on methods (including discussion of the Wnt5a gene regulatory network, the coefficient of determination method for deriving gene regulatory network models, and the partially observable Markov decision process model and its role in modeling intervention planning problems), a main section on the approach taken (including algorithms for solving the intervention planning problems and techniques for representing components of the problems), an empirical evaluation of the intervention planning algorithms on synthetic and the Wnt5a gene regulatory networks, and a conclusion and future directions section. The techniques described present a promising avenue of future research in reasoning algorithms for improved scalability in planning interventions in gene regulatory networks.

Gene regulation plays a central role in the development and functioning of living organisms. Developing a deeper qualitative and quantitative understanding of gene regulation is an important scientific challenge. The switch is commonly used as a paradigm of gene regulation. Verbal descriptions of the structure and functioning of the switch have appeared in biological textbooks. We apply fuzzy modeling to transform one such verbal description into a well-defined mathematical model. The resulting model is a piecewise-quadratic second-order differential equation. It demonstrates functional fidelity with known results while being simple enough to allow a rather detailed analysis. Properties such as the number, location, and domain of attraction of equilibrium points can be studied analytically. Furthermore, the model provides a rigorous explanation for the so-called stability puzzle of the switch.

In this chapter, modeling of GRNs using Petri net theory is considered. It aims at providing a conceptual understanding of Petri nets to enable the reader to explore GRNs applying Petri net modeling and analysis techniques. Starting with an overview on modeling biochemical networks using Petri nets, the state-of-the-art with focus on GRNs is described. Other modeling techniques, for example, hybrid Petri nets are

discussed. Basic concepts of Petri net theory are introduced involving special analysis techniques for modeling biochemical systems, for example, MCT-sets, T-clusters, and Mauritius maps. To illustrate these Petri net concepts, a more complex case study–the gene regulation in Duchenne Muscular Dystrophy–is explained in detail, considering the biological background and the interpretation of analysis results. Considering both, advantages and disadvantages, the chapter demonstrates the usefulness of Petri net modeling, in particular for GRNs.

Preface

For decades, molecular geneticists have been intensively studying the individual genes of various organisms and how these genes influence their phenotypic behavior. Unfortunately, it is usually very difficult, if not impossible, to isolate specific genetic signals for any arbitrary behavioral aspect or trait. The problem is analogous to that of finding a grass skirt in a very large haystack. Even if one locates a plausible-looking bit of grass, until its connections are laboriously traced out, one cannot know if it is part of the skirt or, as is much more likely, just an unrelated piece of straw. As an example, there are over 100 genes that are known to affect flowering time in the model plant *Arabidopsis thaliana*. Together, the interactions of these genes comprise a complex signal processing network that integrates multiple internal and external cues to make one of the most critical decisions in a plant's life cycle–when to reproduce. Yet, all together, these genes comprise only 0.4% of the species' complete gene network.

Recent advances in molecular genetic technologies are beginning to shed light on the complex interplay between genes that elicit phenotypic behavior characteristic of any given organism. Even so, unraveling the specific details about how these genetic pathways interact to regulate development, shape life histories, and respond to environmental cues remains a very daunting task.

A wide variety of models depicting gene-gene interactions, which are commonly referred to as *gene regulatory networks* (GRNs), have been proposed in recent literature. While a GRN must be able to mimic experimentally observed behavior, reproducing complex behaviors accurately may entail computationally prohibitive costs. Under these circumstances, model simplicity is an important trade-off for functional fidelity. Consequently, modeling approaches taken are wide and disparate. Machine learning based GRN models are specifically meant for simplicity and/or algorithmic tractability. They rely heavily on computational learning theory, and usually are used to simulate qualitatively, phenotypic behavior of GRNs. We refer to these as high level models. At the other end are more detailed models that take into account the underlying biochemical processes. These models are capable of reproducing realistic gene expressions with great fidelity.

This book is a collection of articles on the various computational tools that are available to decode, model, and analyze GRNs. It is conveniently organized into separate sections, beginning with an introductory section. Each section contains a handful of chapters written by researchers in the field that focus on a specific computational approach.

SECTION 1: INTRODUCTION

The first section contains two introductory chapters on GRNs. Chapter 1 ("*What are Gene Regulatory Networks*") provides a conceptual framework for GRNs. It shows how the complex nonlinear biochemical processes can be linearized and portrayed as simple graphical models. The nodes of such a network

are either individual genes or groups of functionally related ones. The network can have both directed as well as undirected edges. The chapter also highlights the differences between such networks and two other similar structures, transcriptional regulatory networks and co-expression networks.

The next chapter in this section (Chapter 2) is entitled "*Introduction to Gene Regulatory Networks*" and has a slightly different focus. While introducing the GRN as a graph, it also details further biological insights into the various underlying biochemical processes within GRNs. The chapter also surveys recent advances in array-based technologies that are available to study such processes. Only minimum background in advanced mathematics is assumed here, making the chapter very useful to biologists interested in this subject.

SECTION 2: NETWORK INFERENCE

While the previous section introduces GRNs as graphical structures, the chapters in this section focus on systems identification; they shed light on how GRNs can be reverse engineered from experimental data. While simply arranging genes into various functional units may be accomplished easily through simple statistical means, depicting causality between these units is more challenging.

To varying degrees, all four chapters in this section deal with Bayesian network approach. Bayesian networks, a marriage between graph theory and probability theory, are a high level abstraction of GRNs. An introductory, yet thorough mathematical description of Bayesian networks in provided in Chapter 3 ("*Bayesian Networks for Modeling and Inferring Gene Regulatory Networks*"). This chapter considers both discrete probabilities as well as continuous probability distributions. Dynamic Bayesian networks are taken up briefly to show how cyclic graphs can be modeled. The latter half of the chapter casts the tasks of discovering the structure of the Bayesian network and estimating the parameters of its probability distribution(s) as two aspects of learning. Lastly, it addresses issues relating to assessing the performance of inferred networks.

Chapter 4 ("*Inferring Gene Regulatory Networks from Genetical Genomics Data*") addresses techniques that can be applied to establish causality between the various nodes in a GRN. These techniques are based on the joint analysis of DNA marker and expression as well as DNA sequence information. In addition to Bayesian networks, another modeling approach, statistical equation modeling, is discussed.

Boolean networks are a GRN modeling approach where each gene is associated with a simple logical function. Chapter 5 ("*Inferring Genetic Regulatory Interactions with Bayesian Logic-Based Model*") combines this modeling approach with Bayesian networks. Using simple Boolean semantics to depict underlying interactions among gene products allows for the analysis of larger networks, while the Bayesian framework helps penalize overly complex models. As examples, results of applying this method to data from *S. cerevisiae* and to *Plasmodium falciparum* are provided.

Depicting the dynamic interactions of genes within a network as a set of ordinary differential equations helps preserve biochemical fidelity. Unfortunately, this detailed approach is too complex to be extended beyond a few genes. Chapter 6 ("*A Bayes Regularized Ordinary Differential Equation Model for the Inference of Gene Regulatory Networks*"), makes use of the stochastic nature of GRNs to integrate the differential equation models within a probabilistic network. Bayesian learning is applied to determine the parameters of the differential equation model. The effectiveness of this overall approach is demonstrated by applying it to the yeast cell.

SECTION 3: MODELING METHODS

As noise and delays are intrinsic to biochemical processes, they must be accounted for when dealing with the most detailed differential equation models of GRNs. This issue is addressed in Chapter 7 ("*Computational Approaches for Modeling Intrinsic Noise and Delays in Genetic Regulatory Networks*") and in the following one, Chapter 8 ("*Modeling Gene Regulatory Networks with Delayed Stochastic Dynamics*").

A basic Monte Carlo simulation technique to simulate noisy biochemical reactions, as well as a generalization to include delays, are described in both chapters, although to different ends. Chapter 7 follows this with a study into 'coarse grain' approaches, which reduce computational costs when dealing with larger biochemical systems. The methodology is demonstrated with a few case studies. In contrast, Chapter 8 discusses simulation studies with single genes as well as simple networks of genes. It concludes with a genetic algorithm[1] based simulation to investigate how simple GRNs evolve.

Chapter 9 ("*Nonlinear Stochastic Differential Equations Method for Reverse Engineering of Gene Regulatory Networks*") is a study on how structures of GRNs can be obtained from expression data. It uses stochastic differential equation models, where noise is depicted as a Brownian process. The authors show how regulators for genes are selected using heuristics based on statistical and information theoretic principles, and demonstrate this concept with a few case studies.

The last chapter in this section, Chapter 10 ("*Modelling Gene Regulatory Networks with Computational Intelligence Techniques*") introduces computational intelligence techniques in GRNs with a focus on genetic algorithms. The authors propose the guided genetic algorithm as an optimization method for causal modeling of GRNs. Case studies involving both simulated data as well as real yeast data are described to show how their approach works.

SECTION 4: STRUCTURE AND PARAMETER LEARNING

This section contains a set of chapters that are most directly related to algorithmic approaches for learning structures and parameters of GRNs. It begins with Chapter 11 ("*A Synthesis Method of Gene Regulatory Networks based on Gene Expression by Networking Learning*"), which addresses how GRNs can be modeled to produce oscillatory behavior. This is an important problem as oscillations such as circadian rhythm are routinely observed in gene expression patterns. The chapter proposes a recurrent neural network modeling approach to derive networks of low complexity that can produce desired oscillatory sequences.

Chapter 12 ("*Structural Learning of Genetic Regulatory Networks Based on Prior Biological Knowledge and Microarray Gene Expression Measurements*") is a survey of current methods on Bayesian network models of GRNs. It focuses on structure priors derived from experimental results such as protein-protein interactions, transcription factor binding locations, evolutionary relationships as well as existing literature.

The following chapter, Chapter 13 ("*Problems for Structure Learning: Aggregation and Computational Complexity*") offers a critique on current approaches to inferring model structure using standard machine learning techniques. The authors identify three specific factors in support of their argument: that the methods reported in the literature make use of synthetic as opposed to real data, that they claim success when the actual gene network structure is not known, and that only isolated successes are published.

SECTION 5: ANALYSIS AND COMPLEXITY

Large, heterogeneous datasets arising from a variety of experiments, intricacies involved at various stages of the modeling process, as well as the intrinsically complex nature of the genetic interactions within the organisms themselves–shaped through millenia of evolution–all contribute to models that are often difficult to analyze and comprehend. A collection of articles that address this issue is included in this section.

Chapter 14 ("*Complexity of the BN and the PBN Models of GRNs and Mappings for Complexity Reduction*") is intended to provide a generic framework for model complexity reduction in Boolean and probabilistic Boolean networks. Statistical and information theoretic views of complexity are described. Approaches to map larger GRNs into smaller, more tractable ones, while preserving the overall dynamical behavior, are considered within this scheme.

Chapter 15 ("*Abstraction Methods for Analysis of Gene Regulatory Networks*") also addresses the issue of reducing the complexity in GRNs. It details steps that can be taken to merge similar reactions and eliminate insignificant ones from large-scale models of biochemical reactions. Using these simplifications, models based on chemical kinetics can be abstracted into higher level ones called finite state systems.

Chapter 16 ("*Improved Model Checking Techniques for State Space Analysis of Gene Regulatory Networks*") describes a software tool that applies model checking–a technique used to analyze computer programs–to discrete GRN models. Using this technique, steady state characteristics of the models can be examined. Two case studies, the gene network for cell cycle of yeast, as well as that for wing formation in D. melanogaster, illustrate the effectiveness of this technique.

Chapter 17 ("*Determining the Properties of Gene Regulatory Networks from Expression Data*") shows how topological properties of GRNs can be applied to the practical analysis of experimental gene expression data. Using examples that apply this approach, the authors argue that there is much more to regulation between genes than just transcription factors.

Chapter 18 ("*Generalized Boolean Networks: How Spatial and Temporal Choices Influence Their Dynamics*") relaxes the requirements in random Boolean network models, that genes operate in synchrony and that their connectivity remain fixed. These modifications, it is argued, enable Boolean networks to better capture some characteristics present in gene expression, such as activation sequences in genes and periodic attractors.

SECTION 6: HETEROGENEOUS DATA

Linear programming–a simple technique for the constrained optimization of linear functions–can be used to synthesize GRNs from multiple data sources, as the next two chapters show.

In Chapter 19 ("*A Linear Programming Framework for Inferring Gene Regulatory Network by Integrating Heterogeneous Data*"), the authors use linear differential equation models of GRNs to which matrix decomposition methods and linear programming are applied. Data from heterogeneous sources, such as documented literature, protein-protein interaction data, and so forth are added as constraints. Using this formulation, the authors attempt to obtain robust GRN models that are consistent with multiple datasets.

Chapter 20 ("*Integrating Various Data Sources for Improved Quality in Reverse Engineering of Gene Regulatory Networks*") shows how to reverse engineer large-scale GRNs by integrating various data sources, such as information gleaned by text mining of published research. Using this prior knowledge as

soft evidence, a methodology is proposed to obtain GRN models that can account for large error distributions in microarrays. Simulations with yeast cell data corroborate the effectiveness of this method.

SECTION 7: NETWORK SIMULATION STUDIES

Chapter 21 ("*Dynamic Links and Evolutionary History in Simulated Gene Regulatory Networks*") describes computational studies on the evolution of GRNs. Using evolutionary strategies, an algorithmic approach similar to genetic algorithms, the authors are able to simulate the evolution of GRNs that produce stable multicellular growth. They observe that the evolutionary process favors the appearance of negative feedback in the evolved networks. They hypothesize that this is because negative feedback imparts the network with robustness to potentially deleterious mutations.

A new GRN model that incorporates greater biological detail than traditional methods is outlined in the other simulation study in this section (Chapter 22 "*A Model for a Heterogeneous Genetic Network*"). The authors report computer experiments to generate GRNs using this biologically-motivated approach. They examine the topological features and dynamic behaviors of models obtained in this manner, and provide arguments that such models possess features that correlate well with biological observations.

SECTION 8: OTHER STUDIES

One of the purposes of GRNs is to model cellular dynamics, which are usually characterized by stable attractors. In this context, planned external interventions to redirect these networks from abnormal states (as in with the onset of cancer) to more regular ones is important for many applications, such as prescribing effective drugs. In Chapter 23 ("*Planning Interventions for Gene Regulatory Networks as Partially Observable Markov Decision Processes*"), this intervention problem is modeled as a Markov decision process. Two well known algorithms borrowed for artificial intelligence are proposed to solve the problem.

There are two modes of propagation of a bacterial virus known as the λ phage: direct replication and integration with the host bacterium. The decision concerning which mode to adopt is controlled by a simple GRN called the λ switch. Chapter 24 "*Mathematical Modeling of the λ Switch: A Fuzzy Logic Approach*" uses fuzzy logic to model the switch, making it tractable to mathematical treatment. Using this approach, the chapter suggests explanations for certain behavioral aspects of the λ switch, particularly how the bacterium switches to the direct replication mode of transmission when DNA damage occurs in the host.

Chapter 25, "*Petri Nets and GRN Models*," introduces Petri nets, a graphical modeling approach for modeling GRNs. An introduction to Petri nets as well as related techniques useful in modeling biochemical processes is provided. The application of this approach for the gene regulation in Duchenne muscular dystrophy (DMD) is taken up. An analysis of the results sheds lights on the advantages and disadvantages of the method.

CONCLUSION

This book provides a bird's eye view of the vast range of computational methods used to model GRNs. It contains introductory material and surveys, as well as articles describing in-depth research in various

aspects of GRN modeling. The editors expect it to be useful to researchers in a variety of ways. It can provide a comprehensive overview of artificial intelligence approaches for learning and optimization and their use in gene networks to biologists involved in genetic research. It can assist computer science and artificial intelligence theorists in understanding how their methodologies can be applied to GRN modeling. Although not intended to be a textbook, the book can be of immense use as a reference for students and classroom instructors. As the book would bridge the gap between computer science and genomic research communities, it will be very useful to graduate students considering research in this direction. Finally, this book would be useful to industrial researchers involved in gene regulatory modeling.

Sanjoy Das
Doina Caragea
Stephen M. Welch
William H. Hsu

ADDITIONAL READING

Bansal, M., Gatta, G. D., di Bernardo, D. (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, *22*(7), 815–822.

Bolouri, H. (2008). *Computational modeling of gene regulatory networks: A primer*. World Scientific.

Davidich, M., & Bornholdt, S. (2008). The transition from differential equations to Boolean networks: A case study in simplifying a regulatory network model. *Journal of Theoretical Biology*, *255*(3), 269–277.

Davidson, E. H. (2006). *The regulatory genome: Gene regulatory networks in development and evolution*. Elsevier.

de Jong, H. (2008). Search for steady states of piecewise-linear differential equation models of genetic regulatory networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *5*(2), 208–222.

Grzegorczyk, M., Husmeier, D., Edwards, K. D., Ghazal, P., & Millar, A. J. (2008). Modelling nonstationary gene regulatory processes with a nonhomogeneous Bayesian network and the allocation sampler. *Bioinformatics*, *24*(18), 2071–2078.

Kærn, M., Blake, W. J., & Collins, J. J. (2003). The engineering of gene regulatory networks. *Annual Review of Biomedical Engineering*, 5, 179–206.

Karlebach, G., & Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9, 770–780.

Koduru, P., Dong, Z., Das, S., Welch, S. M., Roe, J., & Charbit, E. (2008). Multi-objective evolutionary-simplex hybrid approach for the optimization of differential equation models of gene networks. *IEEE Transactions on Evolutionary Computation*, *12*(5), 572–590.

Lähdesmäki, H., Hautaniemi, S., Shmulevich, I., & Yli-Harja, O. (2006). Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Processing*, *86*(4), 814–834.

Schlitt T., & Brazma, A. (2007). Current approaches to gene regulatory network modeling. *BMC Bioinformatics*, *8*(Suppl 6), S9.

Welch, S. M., Dong, Z., Roe, J. L., & Das, S. (2005). Flowering time control: Gene network modeling and the link to quantitative genetics. *Australian Journal of Agricultural Research*, 56, 919–936.

Wilczek, A. M., Roe, J., Knapp, M. C., Cooper, M. D., Lopez-Gallego, C., Martin, L. J., Muir, C. D., Sim, S., Walker, A., Anderson, J., Egan, J. F., Moyers, B. T., Petipas, R., Giakountis, A., Charbit, E., Coupland, G., Welch, S. M., & Schmitt, J. (2009). Effects of genetic perturbation on seasonal life history plasticity. *Science*, *323*(5916), 930–934.

ENDNOTE

[1] Genetic algorithms are a class of approaches borrowed from computational intelligence for stochastic optimization. The usage of the word “genetic” does not imply a direct relationship with GRNs, but stems from the fact that these algorithms loosely mimic biological evolution.

Acknowledgment

The editors would like to thank Nancy Williams and Jayme Brown for their kind help and support during the painstaking process of editing this book. They would also like to thank Amity Wilczek for her suggestions on the preface and everyone who participated in the review process. The editors are appreciative of all their insightful comments. Finally, the editors would like to express their gratitude to the authors of the 25 chapters in this book, each of whose contributions has made this book a success.

This work has been supported in part by the U.S. National Science Foundation through Grant No. NSF FIBR 0425759.

Sanjoy Das
Doina Caragea
Stephen M. Welch
William H. Hsu

Section 1
Introduction

Chapter 1
What are Gene Regulatory Networks?

Alberto de la Fuente
CRS4 Bioinformatica, Italy

ABSTRACT

This book deals with algorithms for inferring and analyzing Gene Regulatory Networks using mainly gene expression data. What precisely are the Gene Regulatory Networks that are inferred by such algorithms from this type of data? There is still much confusion in the current literature and it is important to start a book about computational methods for Gene Regulatory Networks with a definition that is as unambiguous as possible. In this chapter, I provide a definition and try to clearly explain what Gene Regulatory Networks are in terms of the underlying biochemical processes. To do the latter in a formal way, I will use a linear approximation to the in general non-linear kinetics underlying interactions in biochemical systems and show how a biochemical system can be 'condensed' into the more compact description of Gene Regulatory Networks. Important differences between the defined Gene Regulatory Networks and other network models for gene regulation, such as Transcriptional Regulatory Networks and Co-Expression Networks, will be highlighted.

INTRODUCTION

Several terms have been used to indicate models of regulatory processes and functional relations between genes, such as Gene Regulatory Networks, Gene Networks, Gene Expression Networks, Co-Expression Networks, Genetic Regulatory Networks, Transcriptional Regulatory Networks and Genetic Interaction Networks. While often used as such in the literature, not all of the above terms are actually synonyms. I therefore will provide a precise definition of the 'Gene Regulatory Network' and point out the essen-

DOI: 10.4018/978-1-60566-685-3.ch001

tial differences with two other network models frequently used for gene regulation, i.e. Transcriptional Regulatory Networks and Co-Expression Networks.

Before a clear definition of Gene Regulatory Networks can be given, we first need to consider the abstract definition of a 'network', also formally called 'graph'. The mathematical theory of graphs is called graph theory (Bollobas, 1998; Erdös & Renyi, 1959), but recent advances in Complex Network Science go beyond graph theory alone and incorporate ideas from physics, sociology and biology (Barabasi & Oltvai, 2004; Dorogovtsev & Mendes, 2003; Newman, 2003; Pieroni et al., 2008; Watts & Strogatz, 1998). Three main types of graphs are essential in the context of Gene Regulatory Networks:

An undirected graph G is an ordered pair G: = (V, U) that is subject to the following conditions:

V is a set, whose elements are called vertices or nodes (the later will be used in the remainder of the chapter) and U is a set of unordered pairs of distinct vertices, called undirected edges, links or lines ('undirected edges' will be used in the remainder of the chapter). For each edge uij = {vi, vj} the nodes vi and vj are said to be connected, linked or adjacent to each other. Undirected graphs can be effectively used to represent the existence of associations or functional relationships (edges) between entities (nodes).

A directed graph or digraph G is an ordered pair G: = (V, D) with V being a set of nodes and D a set of ordered pairs of vertices, called directed edges, arcs, or arrows ('directed edges' will be used in the remainder of the chapter). A directed edge dij = {vi, vj} is considered to be directed from node vi to vj; vj is called the head or target and vi is called the tail or source; vj is said to be a direct successor, or child, of vi, and vi is said to be a direct predecessor, or parent, of vj. If a directed path leads from vi to vj, then vi is said to be an ancestor of vj. Directed graphs can be effectively used to represent causal influences or communication between the nodes.

A mixed graph G is a graph in which some edges may be directed and some may be undirected. It is written as an ordered triple G:= (V, U, D) with V, U, and D defined as above. Directed and undirected graphs are special cases of such mixed graphs. These graphs can thus represent associations as well as causal influences between the nodes. As we will see, Gene Regulatory Networks can most completely be represented as mixed graphs.

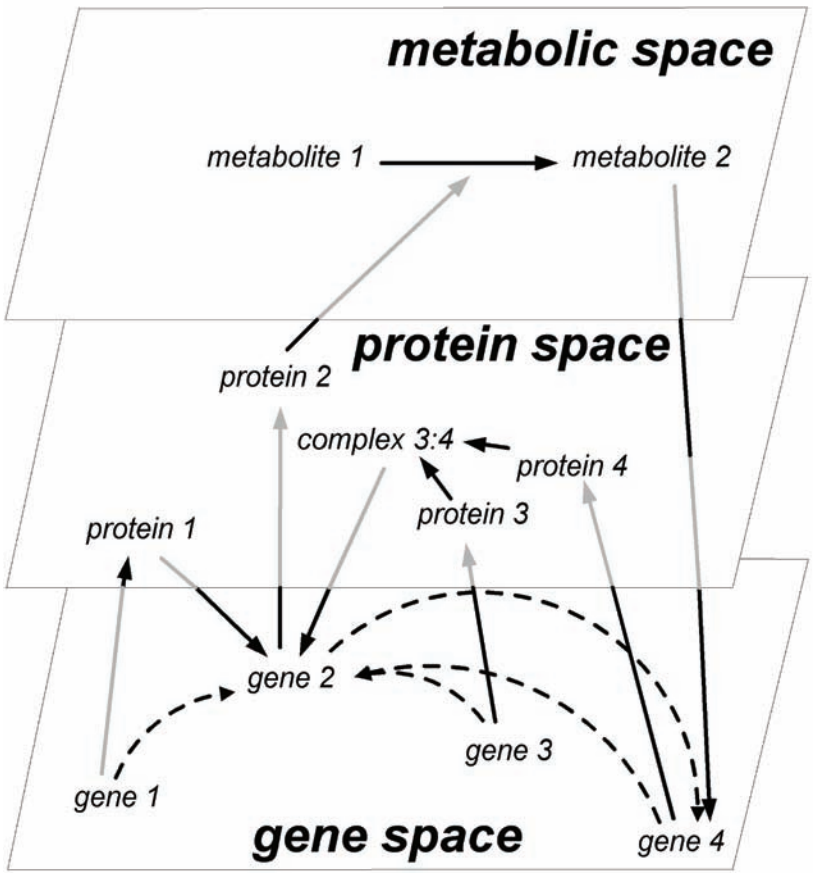
GENE REGULATORY NETWORKS

I start out by giving a possible formal definition for Gene Regulatory Networks. The remainder of the chapter is entirely dedicated to provide a detailed explanation of this definition.

Definition – Gene Regulatory Network (GRN): a Gene Regulatory Network is a mixed graph G:= (V, U, D) over a set V of nodes, corresponding to gene-activities, with unordered pairs U, the undirected edges, and ordered pairs D, the directed edges. A directed edge dij from vi to vj is present iff a causal effect runs from node vi to vj and there exist no nodes or subsets of nodes in V that are intermediating the causal influence (it may be mediated by hidden variables, i.e. variables not in V). An undirected edge uij between nodes vi and vj is present iff gene-activities vi and vj are associated by other means than a direct causal influence, and there exist no nodes or subsets of nodes in V that explain that association (it is caused by a variable hidden to V).

What do the nodes in GRNs precisely represent? The nodes in GRNs are often said to correspond to 'genes'. More precisely, they rather correspond to 'gene-activities' ('gene expression levels' or 'RNA concentrations') as these are the dynamical and quantitative variables that are related by the algorithms discussed in this book. Of course 'gene-activity' could be included in the definition of 'gene'. Therefore, there will be no need to adapt the name 'Gene-activity Regulatory Networks'.

Figure 1. Abstract depiction of cellular physiology. Reprinted with permission from Elsevier from Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene Networks: How to Put the Function in Genomics. In Trends in Biotechnology, 20(11), 6.



What do the edges in GRNs precisely represent? The directed edges in GRNs correspond to causal influences between gene-activities. These could include regulation of transcription by transcription factors, but also less intuitive causal effects between genes involving signal-transduction or metabolism (Figure 2). It is of uttermost importance to realize that when inferring GRNs from gene-expression data alone, the metabolites and proteins act as hidden variables. These variables mediate communication between genes, but since they are not included explicitly in the GRNs, only their effects appear as edges between the observed variables, i.e. the gene-activities. Only cause-effect relations between observed quantities can be established. No matter of how many hidden intermediate causal steps are involved between them, the effects appear to be direct with respect to the set of observed variables. GRNs thus describe communication between genes implicitly including all regulatory processes inside living cells and therefore give a complete description of cellular regulation projected on the gene activities. GRNs are phenomenological, since the mechanisms underlying the edges are generally unknown (yet) and could correspond to complicated paths through proteins and metabolites. However, GRNs are based on a dynamic view of gene regulation: the presence of communication is important, while the precise mechanism of communication is of secondary importance.

Figure 2. The GRN corresponding to the system depicted in figure 1. Reprinted with permission from Elsevier from Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene Networks: How to Put the Function in Genomics. In Trends in Biotechnology, 20(11), 6.

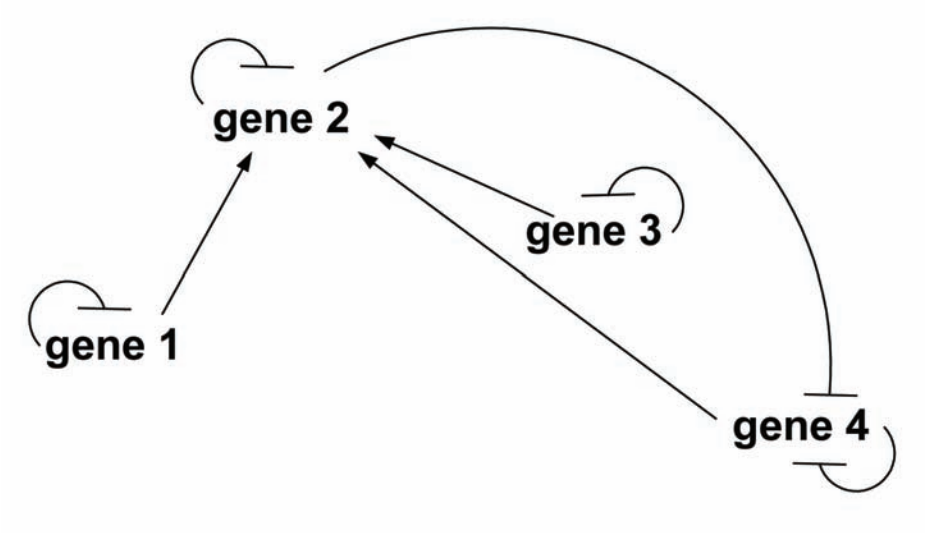


Figure 1 shows a simplified depiction of the biochemistry of living cells conceptually decomposed in three 'spaces' (also referred to as 'levels' in this chapter). Influences between gene-activities, without explicitly taking account for the proteins and metabolites, result from a projection of all regulatory processes on the 'gene space' (Brazhnik, de la Fuente & Mendes, 2002).

Figure 2 shows the GRN resulting from the projection. The influence of gene-activity 1 on gene-activity 2 could have a straightforward interpretation: gene 1 codes for a Transcription factor that regulates gene 2. But an alternative explanation is also possible: protein 1 could modify the rate of gene 1's RNA degradation. The GRN representation doesn't distinguish between the mechanisms as it only accounts for the causal effects: inhibiting a gene's activity could occur through inhibition of transcription or activation of RNA degradation. The effects of gene-activities 3 and 4 are more complicated: their protein products form a complex and then regulate gene 2. The effect gene 2 on gene 4 involves all three levels. Note that the edge from gene-activity 2 to gene-activity 4 will never be present in a Transcriptional Regulatory Network (discussed below), because the protein product of gene 2 does not physically bind to the promoter region of gene 4 to establish its effect. Nevertheless, as we consider only the causal relations between gene-activities, by all means, this effect is considered direct, as the underlying cascade of causality is hidden with respect to the observed quantities.

The undirected edges in GRNs represent 'associations' (for example 'correlations') between gene-activities, due to effects of confounding hidden variables (such as metabolites and proteins). The undirected edges should not be confused with reciprocal effects, i.e. two nodes that are connected by directed edges in both directions. In many studies of complex networks, for example in sociological networks (in which nodes are human individuals and edges represent human interactions such as 'friendships'), the undirected edges are interpreted as such. When two human individuals are friends, information flows in both directions between them (at least it is supposed to be that way!) and in this sense such networks are thus actually directed networks with reciprocal directed edges between each connected pair. Then

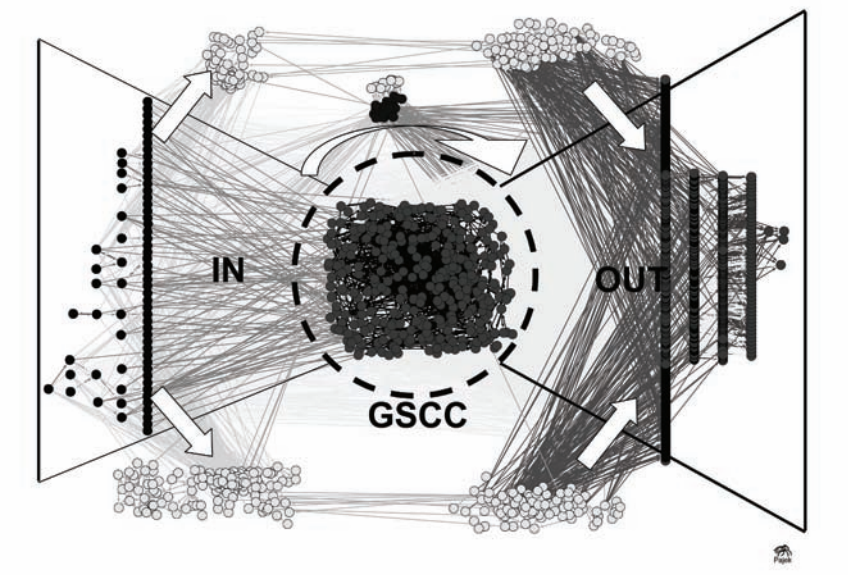
simply out of convenience they are represented as undirected networks. The undirected edges in GRNs can not be interpreted this way: these edges represent associations between pairs of gene-activities that do not correspond to causal influences between the pair. In Genetic Interaction Networks as defined in (Tong et al., 2004) two genes are linked whenever they result in a lethal phenotype when knock-out together, while individual knockouts are viable. The undirected edges in these networks thus reflect a functional similarity between the nodes with respect to a certain phenotype, in contrast to undirected edges in GRNs, which reflect a dynamic association between gene-activities.

As an example, figure 3 shows a partial GRN recently inferred for the yeast S. cerevisiae (Mancosu et al., 2008). The network consists of 4239 nodes and 14,723 directed edges. It is partial in the sense that it lacks the undirected edges that form part of the GRN: only directed edges are presented. The layout is performed according to the networks 'bow tie' structure. Similar structure has been found in metabolic networks of many organisms (Ma & Zeng, 2003) as well as in the World Wide Web (Broder et al., 2000).

In the middle of the network there appears a Giant Strongly Connected Component (GSCC) of 339 genes and 1643 edges. In this component all nodes are connected by cycles. A directed cycle is defined by a directed path starting at a certain node and ending at that same node. The nodes in the IN component (74 nodes and 78 edges) can reach the GSCC through directed paths, but not vice versa. The nodes in the OUT component (3268 nodes and 1559 edges) can be reached from the GSCC but not vice versa. 'Tubes' contain nodes connecting IN to OUT without going through the GSCC. Nodes which are reached from the IN and reach the OUT but which do not belong to any of the aforementioned components are called 'tendrils' (530 nodes and 197 edges). Many edges interface the components: between IN and GSCC 113 edges, GSCC and OUT 9630 edges and between IN and OUT 769 edges.

It is not possible to identify causality from all types of experimental data. In certain cases the algorithms will only be able to produce an undirected network as a final result in which the undirected edges

Figure 3. The bow-tie structure of the yeast GRN. The picture was obtained by combining several layout algorithms implemented in Pajek (Batagelj & Mrvar, 2003). Arrows indicate the direction of the flow of information (taken from (Mancosu et al., 2008)).



could correspond to direct causal influences. Such networks are not GRNs, but rather Co-Expression Networks (CENs).

Co-Expression Networks (CENs)

Similar to GRNs, CENs are inferred from gene expression data. In CENs two genes are connected by an undirected edge if their activities have significant association over a series of gene expression measurements, usually quantified by Pearson correlation (Butte, Tamayo, Slonim, Golub & Kohane, 2000; D'Haeseleer, Liang & Somogyi, 2000), Spearman correlation (D'Haeseleer, Liang & Somogyi, 2000) or Mutual Information (Butte & Kohane, 2000; Steuer, Kurths, Daub, Weise & Selbig, 2002). Again, it is also important to emphasize the difference between GRNs and CENs, since the latter has also been mistakenly called GRNs in the literature by several authors. Gene activities can be correlated due to different causal relationships 1) direct effects 2) indirect effects (correlation is transitive) and 3) confounding. Several algorithms have been proposed to eliminate edges corresponding to 2 and 3 (if the confounding variables are measured), thus resulting in a network which is the undirected version of the GRN (de la Fuente, Bing, Hoeschele & Mendes, 2004; Schäfer & Strimmer, 2005a, 2005b; Veiga, Vicente, Grivet, de la Fuente & Vasconcelos, 2007; Wille & Buhlmann, 2006; Wille et al., 2004).

Still, a correlation does not imply causation and many of the undirected edges may be due to hidden confounding factors. In a later section I will explicitly demonstrate how such edges arise. Only gene expression data obtained through a strategy of 'gene perturbations', or other targeted disturbances to the system, allow for inferring causal relationships. While it has been shown that under certain assumptions it is possible to infer causality without making experimental interventions (Pearl, 2000; Spirtes, Glymour & Scheines, 1993), such assumptions are unfortunately not justified in this context. The strongest assumption is that there are no hidden variables with confounding effects on the observed variables (Spirtes, Glymour & Scheines, 1993). Given the fact that gene-activities are generally the only observed quantities in the data used to infer CENs or GRNs, and that all variables mediating the causal effects between them, i.e. the proteins and metabolites are hidden, such assumption can not be justified under any circumstance. Gene perturbations are thus necessary to infer causality and thus GRNs. Such perturbations could be experimentally created by knocking-out or over-expressing genes (de la Fuente, Brazhnik & Mendes, 2001, 2002; Gardner, di Bernardo, Lorenz & Collins, 2003; Hughes et al., 2000; Mnaimneh et al., 2004; Wagner, 2001), or as will be discussed in other chapters in this book, also natural occurring genetic polymorphisms could be used to infer causal relationships between gene-activities (Bing & Hoeschele, 2005; Liu, de la Fuente & Hoeschele, 2008; Zhu et al., 2004) (see also Liu et al. – this book).

Transcriptional Regulatory Networks (TRNs)

As the name already implies, Transcriptional Regulatory Networks (Guelzim, Bottani, Bourgine & Kepes, 2002; Lee et al., 2002; Luscombe et al., 2004; Shen-Orr, Milo, Mangan & Alon, 2002) only include gene-regulation through transcription, which as we saw is only a small fraction of mechanisms by which the communication between gene-activities occurs. TRNs have directed edges between source and target genes only if it has been experimentally established that the protein product of the source gene physically binds to the promoter region of the target gene and thus potentially regulates transcription, using experimental techniques such as the ChIP-Chip (Buck & Lieb, 2004; Iyer et al., 2001; Lee et al., 2002; Lieb, Liu, Botstein & Brown, 2001; Ren et al., 2000). All edges in TRNs are directed and the only

source nodes are genes coding for Transcription Factors (TFs). TRNs are a mechanistic description of gene regulation with a clear molecular interpretation, straightforwardly connecting to the paradigm of 'molecular biology', while the concept of GRNs considered throughout this book requires one to take the point of view of 'systems biology', i.e. taking a more abstract, but integrated system-wide approach, rather than collecting sets of molecular relationships. Given that GRNs summarize the whole of cellular regulation, to gain insight into the global functional and dynamical organization of gene regulation, GRNs rather than TRNs should be studied.

Can we expect large overlap between experimentally identified GRNs and TRNs of a particular organism? While intuitively one would think so, I claim this is not necessarily the case for the following reasons:

1. **Noise:** First of all, in general there may be mistakes in both networks. GRNs are predominantly based on gene expression data (Brazhnik, de la Fuente & Mendes, 2002; D'Haeseleer, Liang & Somogyi, 2000). TRNs are based on predominantly ChIP-Chip data (Harbison et al., 2004; Lee et al., 2002). Both gene expression data and ChIP-Chip data are plagued by inaccuracies. Gene expression data have several sources of error and ChIP-Chip measurements suffer from a-specific binding. A recent paper showed that TFs bind many sites in the genome; many of which are not believed to be near coding sequences at all (Li et al., 2008). It was also shown that many genes whose promoters were bound were not transcribed in response to the binding event (Li et al., 2008). Furthermore, there is a Multiple Hypothesis Testing (MHT) problem (Storey & Tibshirani, 2003). While many algorithms for GRN inference employ (or at least try to do so) a formal procedure to deal with MHT, most TRNs were obtained using arbitrary p-value thresholds (c.f.Storey & Tibshirani, 2003). Better statistical approaches to obtain TRNs from ChIP-Chip data are in development (Margolin, Palomero, Ferrando, Califano & Stolovitzky, 2007).
2. **Physiologically active regulatory processes:** Edges in TRNs that are not present in GRNs could be explained as follows: to formulate TRNs, the ChIP-Chip experiments are often performed in-vitro after cells have been subjected to many different experimental conditions (Harbison et al., 2004). Thus, the TRN could be expected to nearly completely account for all possible transcriptional regulatory events by the TFs. However, as was shown for the yeast TRN, in each particular physiological state only subsets of these regulatory events are dynamically active (Luscombe et al., 2004). Also, in a recent study, the E. coli TRN was compared to a network obtained through gene expression data measured in many different conditions (Faith et al., 2007). Still, only 10% of the 'known' E. coli transcription regulatory interactions were recovered (Faith et al., 2007), in accordance with the observation that only small parts of TRNs are dynamically active or too weakly active to detect from expression data. It was shown for the yeast TRN that only relatively small parts are active in specific physiological states and that the active sub-networks in those states show widely different topological properties (Luscombe et al., 2004), suggesting that topological analysis of TRNs as a whole is rather meaningless. GRNs inferred in a particular physiologically setting will be entirely active since it is constructed from dynamic information on gene-activities. Therefore, it is justified to explore the whole GRNs for topological features, rather than of subgraphs. It must be stressed that the structure of GRNs are context dependent as well: in different experimental settings (different culture media, temperatures, pH etc.) different causal influences between gene-activities will be physiologically active, leading to a different structure of the inferred GRNs. I expect that the structures of the GRNs obtained for different cell types of a multi-cellular organism can be quite different, both in quantitative as well as in qualitative sense.

3. **Regulation beyond Transcription Factors:** The edges in the GRNs not present in the TRN have a straightforward explanation: the GRN contains much regulation beyond simply transcription factors. Certain processes regulate gene expression independently of transcription, for example regulation through RNA degradation and the small interfering RNAs, which were discovered to play a mayor role in regulation of gene-expression levels (Shimoni et al., 2007). Other processes do involve transcription, but the source nodes are not TFs. For example, genes that code for kinases that activate/inactivate TFs upon phosphorylation will have directed edges to the targets of the TFs. Genes coding for enzymes producing metabolites that in turn activate/inactivate TFs by binding to them, will have directed edges to the targets of the TFs.

Comment on Cyclicity

Cyclic network patterns have been found only rarely in TRNs (Lee et al., 2002; Shen-Orr, Milo, Mangan & Alon, 2002). In the TRN of E. coli from RegulonDB (Gama-Castro et al., 2008; Huerta, Salgado, Thieffry & Collado-Vides, 1998; Salgado et al., 2004; Salgado et al., 2006a; Salgado et al., 2000; Salgado et al., 2001; Salgado et al., 2006b; Salgado et al., 1999) there were no cyclic dependencies at all (Shen-Orr, Milo, Mangan & Alon, 2002). This observation was made in 2002 and since then RegulonDB was subjected to several updates. Still, in current updates of RegulonDB only very few cyclic dependencies are listed. In the TRN studied in (Luscombe et al., 2004) there is a cyclic component involving only 25 nodes. The fact that between genes coding for TFs not much feedback seems to be present does not imply that GRNs are largely acyclic as well. Since GRNs result from a projection of all regulatory processes onto gene space, many cycles can be expected. Indeed the cyclic component of the yeast GRN presented in figure 1 shows a large component of 339 nodes. This component will be responsible for most of the dynamical properties of the whole network. Cyclic dependencies are associated with many (if not all!) fundamental properties of living systems, such as homeostasis, robustness, excitability, multi-stationarity and biological rhythms (e.g. cell cycle, circadian rhythm) (Kauffman, 1969; Noble, 2006; Thieffry & Thomas, 1998; Thomas, 1973; Tyson, Chen & Novak, 2003; von Bertalanffy, 1968; Weiner, 1948; Westerhoff & van Dam, 1987). Again, this emphasizes that TRNs are only representing a part of the global regulatory system, lacking the regulation on the Proteome and Metabolome levels. GRNs, on the other hand, represent the entire global regulatory system, but in a more phenomenological way.

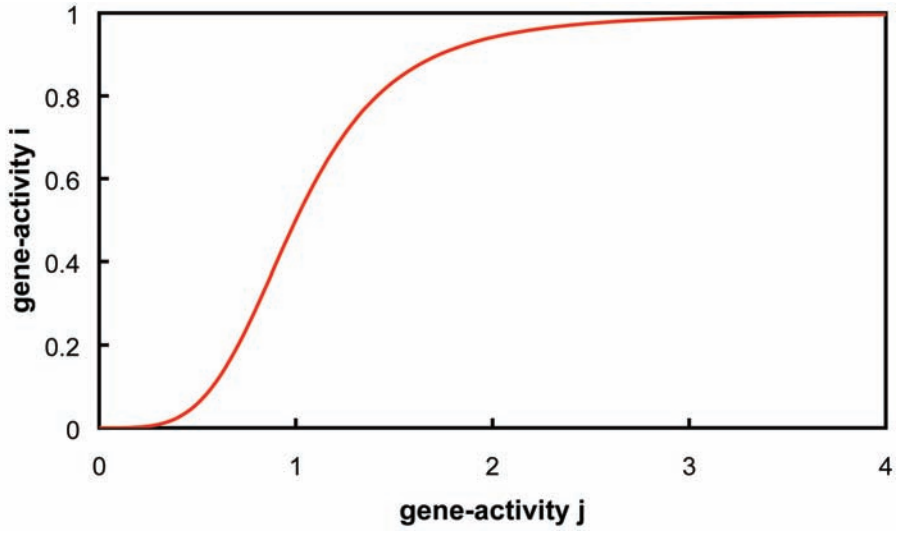
Physiological State Dependent ‘Rewiring’

The structures of GRNs may quantitatively as well as qualitatively depend on the physiological state of the cell. Each of the cell types of a multi-cellular organism can be expected to have GRNs with different structures. Yeast grown in presence of oxygen may have a physiologically active GRN that is different from the physiologically active GRN in anaerobic conditions, etc. How does this ‘rewiring’ happen? One explanation comes from the fact that gene-expression rates are dependent on the activator/inhibitor concentrations in a non-linear (usually hyperbolic or sigmoidal) fashion. Consider the ‘dose-response curve’ given in Figure 4. This example displays the sigmoidal dependence of one gene’s activity on the activity of an activating gene. There are three qualitatively distinct regions in the curve, indicated by the dashed lines. Only in the middle part will the activity of gene i appreciably change upon (small) fluctuations in gene j. In the left and right part the effects are very small, for example, increasing gene-activity j from value 3 to 4 hardly result in any change in gene-activity i. At physiological values of

gene-activity below 0.5 or above 2, gene-activity i will not 'feel' changes in gene-activity j, effectively thus not receiving input from gene-activity j. In each specific physiological state gene-activity j will have different values determined by its inputs in turn. In each physiological state, fluctuations in gene-activity j will 'sample' different parts of this curve, resulting in different strengths of causal influences. This results in quantitative changes in the network structure. If very small effects are ignored (since they are too small of significance to the behavior of the system, or at least can not be determined experimentally) this would translate into qualitative changes in the GRN: edges that appear in one physiological state may not appear in other physiological states.

Several authors (Kauffman, 1969; Thieffry & Thomas, 1998; Thomas, 1973; Wagner, 2001; Yeung, Tegner & Collins, 2002) have argued that GRNs are sparsely connected. However, there are simple arguments that suggest the opposite for GRNs of which I will list a few here. All transcription steps dependent on metabolic energy. Consequently, genes that code for enzymes that have control on the cellular energy level may causally affect all gene-activities. The rates of transcription depend on the concentrations of nucleotides as these are the building blocks of nucleic acids; so all genes coding for enzymes involved in nucleotide synthesis may be inputs of all other genes. Any other genes that affect transcription or RNA degradation, in some general way, will be inputs to all genes. For instance, genes that code for transporters that are responsible for transport of regulating metabolites or proteins into the nucleus. There are many other examples of causal influences that could arise from the complex interplay between the unobserved Proteome and Metabolome and the observed Transcriptome. Since the rate of production of each of the gene-activities competes for the same energy, building blocks, polymerases and transcriptional machinery, an increase in the formation rate of one gene-activity may cause a decrease in all others, implying that GRNs are essentially 'complete graphs', i.e. networks with edges between all pairs of nodes. Whether these numerous potential interactions have a significant magnitude or not is an

Figure 4. Sigmoidal dependence of the value of gene-activity i on the value of gene-activity j. The dashed lines separate regions where gene-activity i is (almost) insensitive to the value of gene-activity j (left and right regions) from the region where gene-activity i is sensitive to the value of gene-activity j (middle region).



open question. Certainly, almost all of these interactions will have small magnitude, as for example in many physiological situations there are plenty of nucleotides such that transcription rates are saturated with them, reducing the related effects to negligible strengths. This situation corresponds to the part of the curve in the third region in figure 4.

'CONDENSING' BIOCHEMISTRY INTO GRNS

Directed Edges

Here I will show how to 'condense' biochemical systems into GRNs in order to clearly demonstrate what the directed edges in GRNs mean in terms of the underlying biochemical processes (de la Fuente & Mendes, 2002). I use the word 'condense', because the GRN is a compact representation of the whole biochemical system; a condensed description of the whole. To this effort is useful to represent a biochemical system as a dynamical system. For each concentration x_i in a biochemical system (metabolites, proteins, gene-activities) a non-linear differential equation can be written to relate its rate of change to a set of parameters k and the set of concentrations x in the system:

$$\frac{dx_i}{dt} = f_i\left(\mathbf{k}, \mathbf{x}\right) \tag{1}$$

For simplicity, I will consider a linearization of the model, but the following reasoning should in principle hold for non-linear systems as well. The linearization describes deviations from a reference state:

$$\Delta\left(\frac{dx_i}{dt}\right) = \sum_j^n a_{ij}\Delta x_j + \Delta u_i \tag{2}$$

The a-coefficients are non-zero iff xj directly affects the rate of change of xi and zero otherwise. These coefficients are elements of a matrix A that represents the wiring structure of the biochemical system. Matrix A is square with dimension n×n, with n the number of variables (e.g. metabolites, proteins and gene-activities) in the biochemical system. An element in row i and column j, i.e. aij, provides the strength by which xj affects xi. If aij is positive, xj activates xi and if negative xj inhibits xi. Matrix A is a so-called weight matrix and corresponds to the Jacobian matrix of the linearized system with elements $\partial\left(dx_i/dt\right)\Big/\partial x_j$, the partial derivatives of rates of changes with respect to the variables. Another matrix representation of networks is the adjacency matrix, which contains simply the number 1 on non-zero positions of A and 0 otherwise. It therefore is a qualitative version of matrix A. Δx_j are the deviations of xj out of the reference state. Δu_i are deviations from the values in the reference state of a rate-parameter that specifically affects dx_i/dt. These deviations can be either seen as experimentally created, i.e. experimental perturbations (interventions), or as spontaneously occurring fluctuations due to 'biological variability': the fact that no repeated observations on the same (or similar) system are identical (even when experimental noise is ignored).

While the study of dynamics in time of GRNs is certainly relevant, especially in studies of organismal development (Bolouri & Davidson, 2003), I will here consider systems in a stable steady state

for the relative simplicity of the following discussion. Note that the main train of thought applies to time-dynamics as well. In a steady state of the biochemical system all activities are constant in time (the time-derivatives are zero) and we can express a relationship between rate-parameter perturbations (fluctuations) and interactions between gene-activities:

$$0 = \sum_{j}^{n} a_{ij} \Delta x_j + \Delta u_i \text{ or } 0 = \sum_{j}^{n} a_{ij} \Delta x_j + \Delta u_i \tag{3}$$

These relations can be written in matrix format

$$\mathbf{AX} = -\mathbf{U} \tag{4}$$

Here A (n×n) is the weight-matrix, U (n×k) is a matrix containing rate fluctuations Δu_{ik}, with elements the deviation of the rate specific to xi in observation k, and X (n×k) is a matrix containing responses (deviations from the reference state) resulting from the fluctuations in U. k is the number of observations made to the system.

Eq. 4 can be written explicitly in terms of the three functional levels of organization of cells, i.e. the Transcriptome, Proteome and Metabolome. One could argue that a 'functional' distinction should not be made, since all bio-molecules, big or small, are 'metabolized' through production and degradation reactions and thus all could be seen as one Metabolome (Cornish-Bowden, Cardenas, Letelier & Soto-Andrade, 2007). Nevertheless, from the point of view of the experimental accessibility of the three levels, it is certainly a useful 'conceptual' distinction. Matrix A can be written in blocks corresponding to the interactions within (diagonal blocks) and between the levels (off-diagonal blocks). Matrices X and U are partitioned accordingly in three separate blocks of rows:

$$\begin{bmatrix} \mathbf{A}_{TT} & \mathbf{A}_{TP} & \mathbf{A}_{TM} \\ \mathbf{A}_{PT} & \mathbf{A}_{PP} & \mathbf{A}_{PM} \\ \mathbf{A}_{MT} & \mathbf{A}_{TP} & \mathbf{A}_{MM} \end{bmatrix} \begin{bmatrix} \mathbf{X}_T \\ \mathbf{X}_P \\ \mathbf{X}_M \end{bmatrix} = - \begin{bmatrix} \mathbf{U}_T \\ \mathbf{U}_P \\ \mathbf{U}_M \end{bmatrix} \tag{5}$$

The subscript T refers 'Transcripts' or 'Transcriptome'(gene-activities), P to 'Proteins' or 'Proteome' and M to 'Metabolites' or 'Metabolome'. Lets take nt as the number of transcripts in the system, np the number of proteins and nm the number of metabolites. The elements of ATT (dimensions nt×nt) represent the effects of the transcript concentrations on the rates of change of transcript concentrations. These effects are mainly due to the degradation rates, since each transcript increases its own degradation rate, transcripts usually do not interfere with the synthesis or degradation of other transcripts (again making the assumption that energy, building bocks and polymerases are not limiting) and transcription is an irreversible process. In the simplest case ATT is merely a lower diagonal matrix with negative numbers: the self-effect due to the enhancement of the degradation rate. Regulation of gene expression by microRNAs will lead to a more complicated form of ATT.

The elements of ATP (nt×np) represent the effects of the protein concentrations on the rates of change of transcript concentrations. RNA-polymerases, Transcription Factors and RNases, for example, are some of the proteins involved in these effects. Also the proteins that make up the spliceosome and proteins that transport mRNA from the nucleus to the cytoplasm will appear in this sub matrix.

ATM (nt×nm) describes the effect of the metabolites on the rate of change of transcript concentrations. Certain metabolites interfere with the transcription of genes by changing the binding affinities of regulating proteins, leading to a change in transcript formation rate. A famous example is tryptophan synthesis in E. coli, in which the trp-operon is inhibited by the concentration of L-tryptophan, the product metabolite of the pathway (Morse, Mosteller & Yanofsky, 1969; Santillan & Mackey, 2001).

APT (np×nt) describes the effects of the transcriptome on the proteome. Since the rate of translation depends on the number of available mRNA molecules each gene-activity positively influences the concentration of the protein it codes for. The columns referring to rRNAs will have positive values in almost every row, since they are part of the ribosomes and thus stimulate the formation rate of all proteins. Also the regulation of translation by microRNAs will give non-zero elements in this sub-matrix.

APP (np×np) contains information of many different types of interaction between proteins. The columns of proteases will have many negative elements; ribosomal proteins will have positive entries in almost all rows. The effects of phosphatases and kinases, and other components of signaling cascades appear in this sub matrix, as well as any other form of protein-protein interaction.

APM (np×nm) shows the effects of metabolites on rate changes in the proteome. Some metabolites interfere with the synthesis or degradation of proteins. For example, protein synthesis and many post-translation modification reactions depend on ATP, GTP and other metabolite concentrations.

AMT (nm×nt) would represent the rare cases of ribozymes catalyzing metabolic reactions, and most entries can be expected to be zero.

AMP (nm×np) mainly contains the effects of metabolic enzymes on the rates of change of substrates and products of the reactions it catalyses. Also contained are the effects of transporters that pump metabolites in and out the cell.

AMM (nm×nm) describes the effects that metabolites have on the rate of change of metabolite concentrations. These are the effects of substrates, products and metabolic modifiers on metabolic reaction rates.

XT (nt×k), XP (np×k), XM (nm×k), UT (nt×k), UP (np×k) and UM (nm×k), with k the total number of measurements made to the system. Experimentally the elements in UT could be accessed by knocking-out genes or over-expressing them (de la Fuente, Brazhnik & Mendes, 2002; Gardner, di Bernardo, Lorenz & Collins, 2003). Experimental perturbations in UP require inhibition/stimulation of for example translation and perturbations in UM could be created by adding inhibitors of metabolic rates.

In the following, the inverse of A is assumed to exist. This is equivalent to assume that the system is present in a structurally stable steady state and that none of the variables can be written as a linear combination of other variables (Heinrich & Schuster, 1996). The responses of the state variables (deviations of the xs from the reference state) towards the perturbations can be written as follows.

$$\begin{bmatrix} \mathbf{X}_T \\ \mathbf{X}_P \\ \mathbf{X}_M \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{TT} & \mathbf{A}_{TP} & \mathbf{A}_{TM} \\ \mathbf{A}_{PT} & \mathbf{A}_{PP} & \mathbf{A}_{PM} \\ \mathbf{A}_{MT} & \mathbf{A}_{MP} & \mathbf{A}_{MM} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{U}_T \\ \mathbf{U}_P \\ \mathbf{U}_M \end{bmatrix} \tag{6}$$

This equation clearly shows how the network of the biochemical system, represented as a weighted matrix, through its inverse transforms the rate-deviations into responses of the concentration of the system variables.

Using the relationship for the inverse of block matrices (Gantmacher, 1960), the inverse of a matrix can be expressed in terms of its blocks (assuming that matrices P and S are non-singular, again related to the structural stability of the sub-systems):

$$\begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Z} & \mathbf{U} \end{bmatrix} \begin{bmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Z} & \mathbf{U} \end{bmatrix} = \begin{bmatrix} \left(\mathbf{P} - \mathbf{Q}\mathbf{S}^{-1}\mathbf{R}\right)^{-1} & -\mathbf{P}^{-1}\mathbf{Q}\left(\mathbf{S} - \mathbf{R}\mathbf{P}^{-1}\mathbf{Q}\right)^{-1} \\ -\mathbf{S}^{-1}\mathbf{R}\left(\mathbf{P} - \mathbf{Q}\mathbf{S}^{-1}\mathbf{R}\right)^{-1} & \left(\mathbf{S} - \mathbf{R}\mathbf{P}^{-1}\mathbf{Q}\right)^{-1} \end{bmatrix}$$

In the present context we are only interested in the top left block, because that is the block that transforms the rate-fluctuations (perturbations) originating in each gene UT into gene-activity responses XT. For the sake of clarity of the following explanation it is assumed that no fluctuations arise or perturbations are made in the Proteome and Metabolome, i.e. UP = 0 and UM = 0. In a later section I will show the implication of fluctuations in those levels separately. Applying the above rule we obtain:

$$\mathbf{X}_T = \left(\mathbf{A}_{TT} - \begin{pmatrix} \mathbf{A}_{TP} & \mathbf{A}_{TM} \end{pmatrix} \begin{pmatrix} \mathbf{A}_{PP} & \mathbf{A}_{PM} \\ \mathbf{A}_{MP} & \mathbf{A}_{MM} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{A}_{PT} \\ \mathbf{A}_{MT} \end{pmatrix} \right)^{-1} \mathbf{U}_T \tag{7}$$

The block rule is applied again on the inverse matrix on the inside and by taking

$$\mathbf{B}_{PP} = \mathbf{A}_{PP} - \mathbf{A}_{PM}\left(\mathbf{A}_{MM}\right)^{-1}\mathbf{A}_{MP}$$
$$\mathbf{B}_{MM} = \mathbf{A}_{MM} - \mathbf{A}_{MP}\left(\mathbf{A}_{PP}\right)^{-1}\mathbf{A}_{PM}$$

we can write XT as:

$$\mathbf{X}_T = \left(\mathbf{A}_{GRN}\right)^{-1}\mathbf{U}_T = \begin{pmatrix} \mathbf{A}_{TT} - \\ \mathbf{A}_{TP}\left(\mathbf{B}_{PP}\right)^{-1}\mathbf{A}_{PT} - \\ \mathbf{A}_{TP}\left(\mathbf{A}_{PP}\right)^{-1}\mathbf{A}_{PM}\left(\mathbf{B}_{MM}\right)^{-1}\mathbf{A}_{MT} + \\ \mathbf{A}_{TM}\left(\mathbf{B}_{MM}\right)^{-1}\mathbf{A}_{MT} - \\ \mathbf{A}_{TM}\left(\mathbf{A}_{MM}\right)^{-1}\mathbf{A}_{MP}\left(\text{'}_{PP}\right)^{-1}\mathbf{A}_{PT} \end{pmatrix}^{-1} \mathbf{U}_T \tag{8}$$

Now we have an expression of AGRN, the weight matrix describing the directed part of the GRN structure: non-zero elements in AGRN correspond to directed edges in the GRN.

$$\begin{aligned}
&\mathbf{A}_{GRN} = \mathbf{A}_{TT} - \\
&\mathbf{A}_{TP}\left(\mathbf{B}_{PP}\right)^{-1}\mathbf{A}_{PT} - \\
&\mathbf{A}_{TP}\left(\mathbf{A}_{PP}\right)^{-1}\mathbf{A}_{PM}\left(\mathbf{B}_{MM}\right)^{-1}\mathbf{A}_{MT} + \\
&\mathbf{A}_{TM}\left(\mathbf{B}_{MM}\right)^{-1}\mathbf{A}_{MT} - \\
&\mathbf{A}_{TM}\left(\mathbf{A}_{MM}\right)^{-1}\mathbf{A}_{MP}\left(\text{'}_{PP}\right)^{-1}\mathbf{A}_{PT}
\end{aligned} \tag{9}$$

The way this equation is presented shows clearly how the communication between genes, given by the weight-matrix $\mathbf{A}_{GRN}$ is composed of several contributions that run through the entire system. $\mathbf{A}_{GRN}$ is then a 'condensed' representation of the whole system. First of all, there is a 'local' effect on the gene-activities, i.e. $\mathbf{A}_{TT}$. Then, influences mediated separately by the Proteome, $\mathbf{A}_{TP}\left(\mathbf{B}_{PP}\right)^{-1}\mathbf{A}_{PT}$, and Metabolome, $\mathbf{A}_{TM}\left(\mathbf{B}_{MM}\right)^{-1}\mathbf{A}_{MT}$ as well as influences through the Proteome and Metabolome, $\mathbf{A}_{TM}\left(\mathbf{A}_{MM}\right)^{-1}\mathbf{A}_{MP}\left(\text{'}_{PP}\right)^{-1}\mathbf{A}_{PT}$ and Metabolome and Proteome, $\mathbf{A}_{TP}\left(\mathbf{A}_{PP}\right)^{-1}\mathbf{A}_{PM}\left(\mathbf{B}_{MM}\right)^{-1}\mathbf{A}_{MT}$. Note that even though I mention that $\mathbf{A}_{TP}\left(\mathbf{B}_{PP}\right)^{-1}\mathbf{A}_{PT}$ and $\mathbf{A}_{TM}\left(\mathbf{B}_{MM}\right)^{-1}\mathbf{A}_{MT}$ are effects that separately run thorugh the Proteome and Metabolome, the presence of the B matrices in these expressions show that the strengths of the influences depend on cyclic communication between the two levels.

To clearly demonstrate the meaning of the rather abstract derivation of $\mathbf{A}_{GRN}$ above I will here consider an example. The example is chosen to be as simple as possible: it concerns two gene-activities communicating through a metabolite (figure 5). Note that synthesis and degradation rates are explicitly included in the depiction, in order to emphasize that the communication occurs through modifying rates.

The whole matrix A for this system reads:

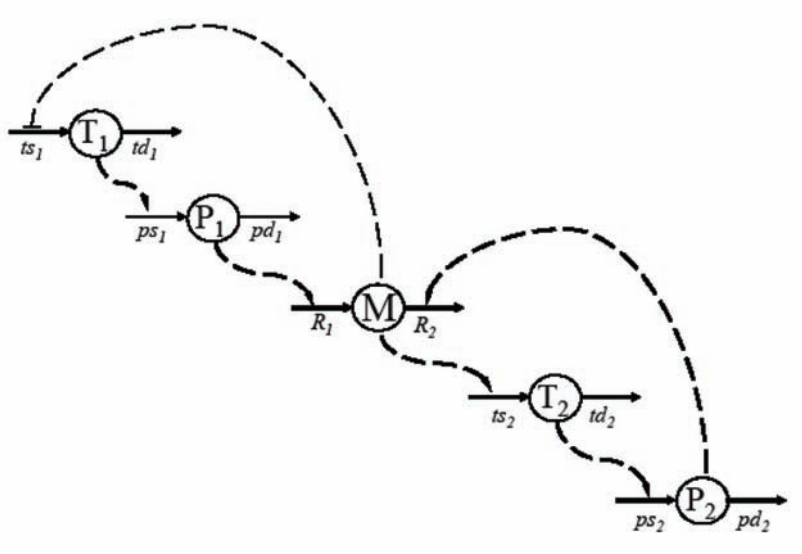
$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{TT} & \mathbf{A}_{TP} & \mathbf{A}_{TM} \\ \mathbf{A}_{PT} & \mathbf{A}_{PP} & \mathbf{A}_{PM} \\ \mathbf{A}_{MT} & \mathbf{A}_{MP} & \mathbf{A}_{MM} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{TT}} & \mathbf{0} & \mathbf{A_{TM}} \\ \mathbf{A_{PT}} & \mathbf{A_{PP}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A_{MP}} & \mathbf{A_{MM}} \end{bmatrix} = \begin{bmatrix} a_{T_1T_1} & 0 & 0 & 0 & a_{T_1M} \\ 0 & a_{T_2T_2} & 0 & 0 & a_{T_2M} \\ a_{P_1T_1} & 0 & a_{P_1P_1} & \mathbf{0} & 0 \\ 0 & a_{P_2T_2} & 0 & a_{P_2P_2} & 0 \\ 0 & 0 & a_{MP_1} & a_{MP_2} & a_{MM} \end{bmatrix} \tag{10}$$

The diagonal elements ('self-effects') appear due to the fact that the degradation rates of each variable depend on their concentrations. Self-effects will always be negative, except if there is an auto-catalytic effect (e.g. a protein that stimulates its own translation) that exceeds the degradation effects in magnitude. When considering the effects between the gene-activities in $\mathbf{A}_{TT}$ we see that each gene-activity only affects itself: without the other system-levels there is no communication between the genes.

By using the expression for $\mathbf{A}_{GRN}$ above, the GRN structure corresponding to the system in figure 5 can be derived. Because $\mathbf{A}_{PM} = \mathbf{0}$ (a matrix full with zeros) note that

$$\begin{aligned}
&\mathbf{B}_{PP} = \mathbf{A}_{PP} \\
&\mathbf{B}_{MM} = \mathbf{A}_{MM}
\end{aligned}$$

Figure 5. A system consisting of two mRNAs, two proteins and a short metabolic pathway of two steps and one metabolic intermediate. There is feedback from the metabolome to the transcriptome. T, P and M represent transcript (mRNA), protein and metabolite, respectively. ts and td stand for rate of transcript synthesis and degradation, respectively; ps and pd stand for rate of protein synthesis and degradation, respectively; and R1 and R2 for metabolic rates. Solid lines indicate mass flow and dashed lines regulatory effects, with arrowheads indicating activation and blunt ends inhibition. Substrates and products of each reaction are not explicitly shown.



This simplification only happens for systems for which there are no direct cycles between the Proteome and Metabolome. The cycles could run indirectly through the Transcriptome as is the case in the current example. Also $\mathbf{A}_{TP} = \mathbf{0}$ and $\mathbf{A}_{TM} = \mathbf{0}$, so that the expression for $\mathbf{A}_{GRN}$ for this system simplifies to:

$$\mathbf{A}_{GRN} = \mathbf{A}_{TT} + \mathbf{A}_{TM}\left(\mathbf{A}_{MM}\right)^{-1}\mathbf{A}_{MP}\left(\mathbf{A}_{PP}\right)^{-1}\mathbf{A}_{PT} \tag{11}$$

Explicitly written out:

$$\mathbf{A}_{GRN} = \begin{pmatrix} a^{*}_{T_1T_1} & a^{*}_{T_1T_2} \\ a^{*}_{T_2T_1} & a^{*}_{T_2T_2} \end{pmatrix} =$$

$$\begin{pmatrix} a_{T_1T_1} & 0 \\ 0 & a_{T_2T_2} \end{pmatrix} + \begin{pmatrix} a_{T_1M} \\ a_{T_2M} \end{pmatrix} \frac{1}{a_{MM}} \begin{pmatrix} a_{MP_1} & a_{MP_2} \end{pmatrix} \begin{pmatrix} \frac{1}{a_{P_1P_1}} & 0 \\ 0 & \frac{1}{a_{P_2P_2}} \end{pmatrix} \begin{pmatrix} a_{P_1T_1} & 0 \\ 0 & a_{P_2T_2} \end{pmatrix} =$$

$$\begin{pmatrix} a_{T_1T_1} + \frac{a_{T_1M}a_{MP_1}a_{P_1T_1}}{a_{MM}a_{P_1P_1}} & \frac{a_{T_1M}a_{MP_2}a_{P_2T_2}}{a_{MM}a_{P_2P_2}} \\ \frac{a_{T_2M}a_{MP_1}a_{P_1T_1}}{a_{MM}a_{P_1P_1}} & a_{T_2T_2} + \frac{a_{T_2M}a_{MP_2}a_{P_2T_2}}{a_{MM}a_{P_2P_2}} \end{pmatrix} \tag{12}$$

The now appearing coefficients a* are the strength of causal influences between genes in the GRN: both gene-activities causally affect each other and themselves. As can be seen in Eq. 12, the weight of these 'phenomenological' coefficients are expressed as a product of a-coefficients along the interaction path (read from right to left), scaled by the self effects of the mediators of the path. The effects of the genes on themselves consist of two parts. One part through degradation (i.e. $a_{T_1T_1}$) and then an effect through the protein and metabolite, which again is expressed as a product of coefficients along the interaction path (read from right to left), scaled by the self effects of the mediators of the path.

Directed edges in GRNs arise through mechanisms such as outline here, i.e. through paths of interactions through the proteome and metabolome. These causal influences are direct in the sense that they are mediated by variables that are not experimentally observed. Only effects mediated by the observed gene-activities are indirect. Consider a path T1 -> P1 -> M1 -> T2 -> P2 -> T3. This path corresponds to an indirect causal influence from T1 to T3, because it crosses T2, which is an observed variable. The GRN will thus not contain a directed edge from T1 to T3, but includes only edges from T1 to T2 and T2 to T3. However, if the transcript T2 is not experimentally measured (or excluded from analysis for some other reason), the GRN resulting from the analysis of the data would include the edge from T1 to T3, because then T2 has the same effect as the proteins and metabolites: it acts as a hidden mediator (de la Fuente, Brazhnik & Mendes, 2001, 2002, 2004).

Undirected Edges

Undirected edges in GRNs arise due to fluctuations in 'confounding' hidden variables. For example, fluctuations in the concentration of a protein that affects two gene-activities will cause the gene-activities to be correlated. Since the protein is not explicitly represented in the GRN, its effect simply remains as an undirected edge representing the association between the gene-activities it causes.

Undirected edges could be represented through for example 'covariances' (or their scaled version 'Pearson correlation'). Covariances can be calculated as follows (assuming that the mean coincides with the reference steady state of Eq. 3 and the fluctuations are random variables identically and independently distributed (i.i.d.) around the mean):

$$\sigma_{ij} = \sum_{k=1}^{n} \Delta x_{ik} \Delta x_{jk} \tag{13}$$

Δx_{ik} and Δx_{jk} are deviations from the mean of gene-activity xi and xj, respectively, in observation k. n is the total number of observations.

Using the matrix equation (Eq. 4), the co-variance matrix can be expressed as

$$\boldsymbol{\Sigma}_X = \mathbf{X}\mathbf{X}^T = \mathbf{A}^{-1}\mathbf{U}\mathbf{U}^T \left(\mathbf{A}^{-1}\right)^T = \mathbf{A}^{-1}\boldsymbol{\Sigma}_U \left(\mathbf{A}^{-1}\right)^T \tag{14}$$

The covariances depend in a complicated way on the structure of A, i.e. through its inverse. This is the reason that covariances and correlations are known to be transitive: if A affects B and B affects C there will be correlation between A and C. In addition, if B affects A and B affects C there will be correlation between A and C. $\boldsymbol{\Sigma}_U$ is a covariance matrix containing covariances between rate-fluctuations.

It is often assumed to be diagonal, i.e. all rate-fluctuations are independent and it only contains the rate-fluctuation variances (Bollen, 1989). For the Metabolome is not completely a justifiable assumption, since metabolites are coupled by fluxes and a fluctuation in a conversion rate between substrate and product will directly cause a dependent fluctuation in both metabolites (Camacho, de la Fuente & Mendes, 2005): nevertheless the assumption is made for simplicity of the coming discussion.

The 'inverse covariance matrix' (Dempster, 1972; Edwards, 1995) has a simpler relationship to the structure of the system A.

$$\mathbf{\Omega}_{\mathbf{X}} = \mathbf{\Sigma}_{\mathbf{X}}{}^{-1} = \mathbf{A}^{T}\mathbf{\Sigma}_{U}{}^{-1}\mathbf{A} \tag{15}$$

The inverse covariance matrix holds partial variances on its diagonal and partial covariances in its off-diagonal elements (Schäfer & Strimmer, 2005a). The interpretation of the partial covariances is the covariance that remains after conditioning on all other variables. If $\mathbf{\Sigma}_{U}$ is diagonal, there is a clear relationship between the inverse of the co-variance matrix and to the structure of the system: A pre-multiplied by its transpose scaled by the variance of the fluctuations. The matrix $\mathbf{A}^{T}\mathbf{A}$ corresponds to the 'moral graph' of the network corresponding to A. The moral graph is an undirected graph and can straightforwardly be obtained from the original graph by 'undirecting' the directed edges and placing an undirected edge between any pair of nodes that share the same target (Cowell, Dawid, Lauritzen & Spiegelhalter, 1999). So, when a series of i.i.d. observations on all variables in the system is made, one can estimate the co-variance matrix using for example, shrinkage estimation (Schäfer & Strimmer, 2005b), take the inverse, decide on a threshold for non-zero elements and obtain a matrix corresponding to an undirected version of the network with certain additional edges (between the parent-nodes of each child-node). These latter edges are unwanted and could in principle be removed by low-order partial correlation tests (de la Fuente, Bing, Hoeschele & Mendes, 2004; Pearl, 2000).

However, in general not the whole system is observed. The data considered pre-dominantly in this book contains observations on only the gene-activities. Again, consider the subdivision of the whole system into the Transcriptome, Proteome and Metabolome. The covariance matrix contains diagonal blocks with covariances between variables in the same level and off-diagonal blocks with covariances between variables across levels.

$$\mathbf{\Sigma}_{X} = \begin{bmatrix} \mathbf{\Sigma}_{TT} & \mathbf{\Sigma}_{TP} & \mathbf{\Sigma}_{TM} \\ \mathbf{\Sigma}_{PT} & \mathbf{\Sigma}_{PP} & \mathbf{\Sigma}_{PM} \\ \mathbf{\Sigma}_{MT} & \mathbf{\Sigma}_{MP} & \mathbf{\Sigma}_{MM} \end{bmatrix} = \begin{bmatrix} \mathbf{\Omega}_{TT} & \mathbf{\Omega}_{TP} & \mathbf{\Omega}_{TM} \\ \mathbf{\Omega}_{PT} & \mathbf{\Omega}_{PP} & \mathbf{\Omega}_{PM} \\ \mathbf{\Omega}_{MT} & \mathbf{\Omega}_{MP} & \mathbf{\Omega}_{MM} \end{bmatrix}^{-1} \tag{16}$$

Of all the sub-matrices only can be estimated from gene-expression measurements. Therefore, it is relevant to show what in theory is obtained by taking its inverse. Again, the block inverse relationship is used.

$$\boldsymbol{\Sigma}_{TT}^{-1} = \boldsymbol{\Omega}_{TT} - \begin{pmatrix}\boldsymbol{\Omega}_{TP} & \boldsymbol{\Omega}_{TM}\end{pmatrix}\begin{pmatrix}\boldsymbol{\Omega}_{PP} & \boldsymbol{\Omega}_{PM} \\ \boldsymbol{\Omega}_{MP} & \boldsymbol{\Omega}_{MM}\end{pmatrix}^{-1}\begin{pmatrix}\boldsymbol{\Omega}_{PT} \\ \boldsymbol{\Omega}_{MT}\end{pmatrix} =$$
$$\boldsymbol{\Omega}_{TT} -$$
$$\boldsymbol{\Omega}_{TP}\left(\mathbf{C}_{PP}\right)^{-1}\boldsymbol{\Omega}_{PT} -$$
$$\boldsymbol{\Omega}_{TP}\left(\boldsymbol{\Omega}_{PP}\right)^{-1}\boldsymbol{\Omega}_{PM}\left(\mathbf{C}_{MM}\right)^{-1}\boldsymbol{\Omega}_{MT} +$$
$$\boldsymbol{\Omega}_{TM}\left(\mathbf{C}_{MM}\right)^{-1}\boldsymbol{\Omega}_{MT} -$$
$$\boldsymbol{\Omega}_{TM}\left(\boldsymbol{\Omega}_{MM}\right)^{-1}\boldsymbol{\Omega}_{MP}\left(\mathbf{C}_{PP}\right)^{-1}\boldsymbol{\Omega}_{PT} \tag{17}$$

Here

$$\mathbf{C}_{PP} = \boldsymbol{\Omega}_{PP} - \boldsymbol{\Omega}_{PM}\left(\boldsymbol{\Omega}_{MM}\right)^{-1}\boldsymbol{\Omega}_{MP}$$
$$\mathbf{C}_{MM} = \boldsymbol{\Omega}_{MM} - \boldsymbol{\Omega}_{MP}\left(\boldsymbol{\Omega}_{PP}\right)^{-1}\boldsymbol{\Omega}_{PM}$$

Again, like for the causal representation was shown that communication between genes arose through paths through the Proteome and Metabolome, here it is seen that the edges between gene-activities in the moral graph of the GRN arise due to partial covariances of the gene-activities with proteins and metabolites. These covariances may arise through paths of communication between genes-activities, as described above, but also occur due to confounding by hidden variables of the other levels. If all variables would be measured and the complete covariance matrix is considered, these covariances would drop out. But not incorporating this information (simply because the data is usually not available) will result in undirected edges in the inferred GRNs.

Consider again the simple system in Figure 5. For the following demonstration a modification is considered in which the proteins do not affect the metabolic rate, i.e. $\mathbf{A}_{MP} = \mathbf{0}$. Without these effects there is no causal influence between the gene-activities. The metabolite still regulates both gene-activities and it will be shown that fluctuations in the metabolite will give rise to an undirected edge between the two genes. For clarity of the following demonstration $\boldsymbol{\Sigma}_U$ is set to the identity matrix. A diagonal $\boldsymbol{\Sigma}_U$ with different values will merely result in scaled coefficients. The general case of having a non-diagonal $\boldsymbol{\Sigma}_U$ will result in having yet additional covariances obscuring the simple relationship between the network structure and the inverse covariance matrix.

The A matrix of the system under consideration reads:

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}_{TT} & \mathbf{A}_{TP} & \mathbf{A}_{TM} \\ \mathbf{A}_{PT} & \mathbf{A}_{PP} & \mathbf{A}_{PM} \\ \mathbf{A}_{MT} & \mathbf{A}_{MP} & \mathbf{A}_{MM}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{TT} & \mathbf{0} & \mathbf{A}_{TM} \\ \mathbf{A}_{PT} & \mathbf{A}_{PP} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{MM}\end{bmatrix} = \begin{bmatrix}a_{T_1T_1} & 0 & 0 & 0 & a_{T_1M} \\ 0 & a_{T_2T_2} & 0 & 0 & a_{T_2M} \\ a_{P_1T_1} & 0 & a_{P_1P_1} & 0 & 0 \\ 0 & a_{P_2T_2} & 0 & a_{P_2P_2} & 0 \\ 0 & 0 & 0 & 0 & a_{MM}\end{bmatrix} \tag{18}$$

The corresponding concentration matrix reads:

$$\mathbf{\Omega}_X = \mathbf{A}^T\mathbf{A} = \begin{bmatrix} \omega_{T_1} & 0 & a_{P_1P_1}a_{P_1T_1} & 0 & a_{T_1T_1}a_{T_1M} \\ 0 & \omega_{T_2} & 0 & a_{P_2P_2}a_{P_2T_2} & a_{T_2T_2}a_{T_2M} \\ a_{P_1P_1}a_{P_1T_1} & 0 & \omega_{P_1} & 0 & 0 \\ 0 & a_{P_2P_2}a_{P_2T_2} & 0 & \omega_{P_2} & 0 \\ 0 & 0 & 0 & 0 & \omega_M \end{bmatrix} = \begin{bmatrix} \mathbf{\Omega}_{TT} & \mathbf{\Omega}_{TP} & \mathbf{\Omega}_{TM} \\ \mathbf{\Omega}_{PT} & \mathbf{\Omega}_{PP} & \mathbf{0} \\ \mathbf{\Omega}_{MT} & \mathbf{0} & \mathbf{\Omega}_{MM} \end{bmatrix} \tag{19}$$

Note that the partial variances are not explicitly written out, since we are only interested in the off-diagonal elements. No edges between the Metabolome and the Proteome are present in the concentration matrix, but their covariance will be non-zero, as there is a causal influence from the metabolite to the proteins through the transcripts. $\mathbf{\Sigma}_U$ for this example system indeed is fully nonzero (result not shown): all variables are correlated to some extend.

For this system the general equation reduces to:

$$\mathbf{\Sigma}_{TT}{}^{-1} = \mathbf{\Omega}_{TT} - \mathbf{\Omega}_{TP}\left(\mathbf{\Omega}_{PP}\right)^{-1}\mathbf{\Omega}_{PT} + \mathbf{\Omega}_{TM}\left(\mathbf{\Omega}_{MM}\right)^{-1}\mathbf{\Omega}_{MT} \tag{20}$$

Written out explicitly:

$$\mathbf{\Sigma}_{TT}{}^{-1} = \begin{pmatrix} \omega_{T_1} & 0 \\ 0 & \omega_{T_2} \end{pmatrix} - \begin{pmatrix} a_{P_1P_1}a_{P_1T_1} & 0 \\ 0 & a_{P_2P_2}a_{P_2T_2} \end{pmatrix} \begin{pmatrix} \frac{1}{\omega_{P_1}} & 0 \\ 0 & \frac{1}{\omega_{P_2}} \end{pmatrix} \begin{pmatrix} a_{P_1P_1}a_{P_1T_1} & 0 \\ 0 & a_{P_2P_2}a_{P_2T_2} \end{pmatrix} + \begin{pmatrix} a_{T_1T_1}a_{T_1M} \\ a_{T_2T_2}a_{T_2M} \end{pmatrix} \frac{1}{\omega_M} \begin{pmatrix} a_{T_1T_1}a_{T_1M} & a_{T_2T_2}a_{T_2M} \end{pmatrix} =$$

$$\begin{pmatrix} \omega_{T_1}{}^* & \frac{a_{T_1T_1}a_{T_1M}a_{T_2T_2}a_{T_2M}}{\omega_M} \\ \frac{a_{T_1T_1}a_{T_1M}a_{T_2T_2}a_{T_2M}}{\omega_M} & \omega_{T_2}{}^* \end{pmatrix} \tag{21}$$

Again the diagonal terms are not written out. Note that now off-diagonal elements are observed in the concentration matrix although there are no causal effects between the genes: both genes are dependent on the metabolite and the corresponding element in the inverse of the covariance matrix will be non-zero due to this effect. Since the metabolite is hidden in the analysis of gene-expression data the undirected edge will appear in the GRN: none of the observed variables, i.e. gene-activities, can explain this correlation. The covariance with the Proteome only affects the partial variances of the gene-activities. The demonstration here confirms what already was intuitively clear: hidden variables with confounding effects will create associations between the observed associated.

CONCLUSION

In this chapter I tried to convey several messages. First, I gave a formal definition of GRNs. Second, I pointed out the conceptual differences between GRNs, CENs and TRNs. Contrary to what is often believed, GRNs and TRNs are conceptually very different. Directed edges in GRNs do no necessarily

originate in Transcription Factors. The directed edges in GRNs correspond to causal paths of influence through the Proteome and Metabolome, which are usually not considered in data used to infer GRNs. The undirected edges in GRNs correspond to confounding influences on gene-activities by the Proteome and Metabolome. Again, since metabolites and proteins are hidden to the analysis methods, such edges can not be removed, since none of the observed gene-activities can explain these. Third, I argued that the directed edges in GRNs can only be inferred through perturbation data. The assumptions needed to be able to infer causality from non-perturbation data (also called 'non-experimental data' or 'observational data') can never be justified for biochemical systems: too many hidden variables are into play and could act as hidden confounding disturbances with respect to the observed part of the system, i.e. the gene-activities.

Perturbation data is required to establish the directed edges in GRNs. Several methods have been proposed for that purpose based on experimental perturbations (de la Fuente, Brazhnik & Mendes, 2001, 2002; Gardner, di Bernardo, Lorenz & Collins, 2003; Wagner, 2001) and using naturally occurring genetic perturbations, i.e. polymorphisms in genes (Bing & Hoeschele, 2005; Liu, de la Fuente & Hoeschele, 2008; Zhu et al., 2004) (see also Liu et al. – this book). Observational (i.d.d.) data allows for inferring CENs and the undirected edges in GRNs. Several methods have been proposed for this purpose too (de la Fuente, Bing, Hoeschele & Mendes, 2004; Schäfer & Strimmer, 2005a, 2005b; Veiga, Vicente, Grivet, de la Fuente & Vasconcelos, 2007; Wille & Buhlmann, 2006; Wille et al., 2004). A complete GRN is the superposition of the CEN and the collection of directed edges. Such superposition may lead to a GRN with many pairs with both directed and undirected edges. In that case the undirected edge could be dropped, assuming that the undirected edge is caused by the causal influence represented by the directed edge. This is not necessarily a correct assumption: there could be a causal influence between gene-activities and in addition a confounding effect by a hidden variable. It seems to me that this situation is impossible to recognize by analyzing gene-expression data alone, making the above assumption the only alternative.

As shown throughout this chapter, GRNs are rather abstract networks. In contrast to TRNs there is not simple way to associate a clear molecular mechanism to the edges. Nevertheless, since the GRN is a projection of all regulation occurring in the biochemical system it is a complete description of the system in terms of communication and associations between the genes. Given that GRNs summarize the whole of cellular regulation, to gain insight into functional dynamical organization of genetic regulation, GRNs rather than TRNs should be studied. Recent papers indeed show that profound biological insight can be obtained by studying GRNs (Bystrykh et al., 2005; Keurentjes et al., 2007; Mehrabian et al., 2005; Schadt et al., 2005). It is therefore an important goal to infer and analyze these networks, emphasizing the need for books like this one, on computational methods for Gene Regulatory Networks.

REFERENCES

Barabasi, A. L., & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews. Genetics*, *5*(2), 101–113. doi:10.1038/nrg1272

Batagelj, V., & Mrvar, A. (2003). Pajek-analysis and visualization of large networks. In M. Jünger & P. Mutzel (Eds.), Graph drawing software (pp. 77-103). Springer.

Bing, N., & Hoeschele, I. (2005). Genetical genomics analysis of a yeast segregant population for transcription network inference. *Genetics*, *170*(2), 533–542. doi:10.1534/genetics.105.041103

Bollen, K. (1989). Structural equations with latent variables. Wiley-Interscience.

Bollobas, B. (1998). Modern graph theory. Springer.

Bolouri, H., & Davidson, E. H. (2003). Transcriptional regulatory cascades in development: Initial rates, not steady state, determine network kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(16), 9371–9376. doi:10.1073/pnas.1533293100

Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene networks: How to put the function in genomics. *Trends in Biotechnology*, *20*(11), 467–472. doi:10.1016/S0167-7799(02)02053-X

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. (2000). Graph structure in the Web. Computer Networks, 33(1-6), 309-320.

Buck, M. J., & Lieb, J. D. (2004). ChIP-chip: Considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, *83*(3), 349–360. doi:10.1016/j.ygeno.2003.11.004

Butte, A. J., & Kohane, I. S. (2000). Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 418–429.

Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R., & Kohane, I. S. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(22), 12182–12186. doi:10.1073/pnas.220392197

Bystrykh, L., Weersing, E., Dontje, B., Sutton, S., Pletcher, M. T., & Wiltshire, T. (2005). Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics.' . *Nature Genetics*, *37*(3), 225–232. doi:10.1038/ng1497

Camacho, D., de la Fuente, A., & Mendes, P. (2005). The origin of correlations in metabolomic data. *Metabolomics*, *1*(1), 53–63. doi:10.1007/s11306-005-1107-3

Cornish-Bowden, A., Cardenas, M. L., Letelier, J. C., & Soto-Andrade, J. (2007). Beyond reductionism: Metabolic circularity as a guiding vision for a real biology of systems. *Proteomics*, *7*(6), 839–845. doi:10.1002/pmic.200600431

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). Probabilistic networks and expert systems. New York: Springer-Verlag.

D'Haeseleer, P., Liang, S., & Somogyi, R. (2000). Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics (Oxford, England)*, *16*(8), 707–726. doi:10.1093/bioinformatics/16.8.707

de la Fuente, A., Bing, N., Hoeschele, I., & Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics (Oxford, England)*, *20*, 3565–3574. doi:10.1093/bioinformatics/bth445

de la Fuente, A., Brazhnik, P., & Mendes, P. (2001). A quantitative method for reverse engineering gene networks from microarray experiments using regulatory strengths. Paper presented at the 2nd Int. Conf. Syst. Biol., California Institute of Technology, Pasadena, CA.

de la Fuente, A., Brazhnik, P., & Mendes, P. (2002). Linking the genes: Inferring quantitative gene networks from microarray data. *Trends in Genetics*, *18*(8), 395–398. doi:10.1016/S0168-9525(02)02692-6

de la Fuente, A., Brazhnik, P., & Mendes, P. (2004). Regulatory strength analysis for inferring gene networks. In K. B. N. & W. H. V. (Eds.), Metabolic engineering in the post genomic era (pp. 107-137). Wymondham, UK: Horizon Bioscience.

de la Fuente, A., & Mendes, P. (2002). Quantifying gene networks with regulatory strengths. *Molecular Biology Reports*, *29*(1-2), 73–77. doi:10.1023/A:1020310504986

Dempster, A. P. (1972). Covariance selection. *Biometrics*, *28*, 157–175. doi:10.2307/2528966

Dorogovtsev, S. N., & Mendes, J. F. F. (2003). Evolution of networks: From biological networks to the Internet and WWW. Oxford: Oxford University Press.

Edwards, D. (1995). Introduction to graphical modelling. Springer-Verlag.

Erdös, P., & Renyi, A. (1959). On random graphs. *Publ. Math. Debrecen*, *6*, 290–297.

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., & Cottarel, G. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, *5*(1), e8. doi:10.1371/journal.pbio.0050008

Gama-Castro, S., Jimenez-Jacinto, V., Peralta-Gil, M., Santos-Zavaleta, A., Penaloza-Spinola, M. I., & Contreras-Moreira, B. (2008). RegulonDB (version 6.0): Gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Research*, *36*(Database issue), D120–D124. doi:10.1093/nar/gkm994

Gantmacher, F. R. (1960). The theory of matrices (Vol. II). New York: Chelsea Publishing Company.

Gardner, T., di Bernardo, D., Lorenz, D., & Collins, J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *301*(5629), 102–105. doi:10.1126/science.1081900

Guelzim, N., Bottani, S., Bourgine, P., & Kepes, F. (2002). Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics*, *31*(1), 60–63. doi:10.1038/ng873

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., & Danford, T. W. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, *431*(7004), 99–104. doi:10.1038/nature02800

Heinrich, R., & Schuster, S. (1996). The regulation of cellular systems. New York: Chapman + Hall.

Huerta, A. M., Salgado, H., Thieffry, D., & Collado-Vides, J. (1998). RegulonDB: A database on transcriptional regulation in Escherichia coli. *Nucleic Acids Research*, *26*(1), 55–59. doi:10.1093/nar/26.1.55

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., & Armour, C. D. (2000). Functional discovery via a compendium of expression profiles. *Cell*, *102*(1), 109–126. doi:10.1016/S0092-8674(00)00015-5

Iyer, V. R., Horak, C. E., Scafe, C. S., Botstein, D., Snyder, M., & Brown, P. O. (2001). Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, *409*(6819), 533–538. doi:10.1038/35054095

Kauffman, S. (1969). Homeostasis and differentiation in random genetic control networks. *Nature*, *224*(215), 177–178. doi:10.1038/224177a0

Keurentjes, J. J., Fu, J., Terpstra, I. R., Garcia, J. M., van den Ackerveken, G., & Snoek, L. B. (2007). Regulatory network construction in Arabidopsis by using genome-wide gene expression quantitative trait loci. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(5), 1708–1713. doi:10.1073/pnas.0610429104

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., & Gerber, G. K. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, *298*(5594), 799–804. doi:10.1126/science.1075090

Li, X. Y., MacArthur, S., Bourgon, R., Nix, D., Pollard, D. A., & Iyer, V. N. (2008). Transcription factors bind thousands of active and inactive regions in the Drosophila blastoderm. *PLoS Biology*, *6*(2), e27. doi:10.1371/journal.pbio.0060027

Lieb, J. D., Liu, X., Botstein, D., & Brown, P. O. (2001). Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association. *Nature Genetics*, *28*(4), 327–334. doi:10.1038/ng569

Liu, B., de la Fuente, A., & Hoeschele, I. (2008). Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics*, *178*, 1763–1776. doi:10.1534/genetics.107.080069

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A., & Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, *431*(7006), 308–312. doi:10.1038/nature02782

Ma, H. W., & Zeng, A. P. (2003). The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics (Oxford, England)*, *19*(11), 1423–1430. doi:10.1093/bioinformatics/btg177

Mancosu, G., Pieroni, E., Maggio, F., Fotia, G., Liu, B., Hoeschele, I., et al. (2008). Deciphering a genome-wide yeast gene network. Submitted.

Margolin, A. A., Palomero, T., Ferrando, A. A., Califano, A., & Stolovitzky, G. (2007). ChIP-on-chip significance analysis reveals ubiquitous transcription factor binding. *BMC Bioinformatics*, *8*(Suppl 8), S2. doi:10.1186/1471-2105-8-S8-S2

Mehrabian, M., Allayee, H., Stockton, J., Lum, P. Y., Drake, T. A., & Castellani, L. W. (2005). Integrating genotypic and expression data in a segregating mouse population to identify 5-lipoxygenase as a susceptibility gene for obesity and bone traits. *Nature Genetics*, *37*(11), 1224–1233. doi:10.1038/ng1619

Mnaimneh, S., Davierwala, A. P., Haynes, J., Moffat, J., Peng, W. T., & Zhang, W. (2004). Exploration of essential gene functions via titratable promoter alleles. *Cell*, *118*(1), 31–44. doi:10.1016/j.cell.2004.06.013

Morse, D. E., Mosteller, R. D., & Yanofsky, C. (1969). Dynamics of synthesis, translation, and degradation of trp operon messenger RNA in E. coli. *Cold Spring Harbor Symposia on Quantitative Biology*, *34*, 725–740.

Newman, M. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256. doi:10.1137/S003614450342480

Noble, D. (2006). The music of life. Oxford.

Pearl, J. (2000). Causality: Models, reasoning, and inference: Cambridge University Press.

Pieroni, E., de la Fuente van Bentem, S., Mancosu, G., Capobianco, E., Hirt, H., & de la Fuente, A. (2008). Protein networking: Insights into global functional organization of proteomes. *Proteomics*, *8*(4), 799–816. doi:10.1002/pmic.200700767

Ren, B., Robert, F., Wyrick, J. J., Aparicio, O., Jennings, E. G., & Simon, I. (2000). Genome-wide location and function of DNA binding proteins. *Science*, *290*(5500), 2306–2309. doi:10.1126/science.290.5500.2306

Salgado, H., Gama-Castro, S., Martinez-Antonio, A., Diaz-Peredo, E., Sanchez-Solano, F., & Peralta-Gil, M. (2004). RegulonDB (version 4.0): Transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12. *Nucleic Acids Research*, *32*(Database issue), D303–D306. doi:10.1093/nar/gkh140

Salgado, H., Gama-Castro, S., Peralta-Gil, M., Diaz-Peredo, E., Sanchez-Solano, F., & Santos-Zavaleta, A. (2006a). RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research*, *34*(Database issue), D394–D397. doi:10.1093/nar/gkj156

Salgado, H., Santos, A., Garza-Ramos, U., van Helden, J., Diaz, E., & Collado-Vides, J. (1999). RegulonDB (version 2.0): A database on transcriptional regulation in Escherichia coli. *Nucleic Acids Research*, *27*(1), 59–60. doi:10.1093/nar/27.1.59

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millan-Zarate, D., Blattner, F. R., & Collado-Vides, J. (2000). RegulonDB (version 3.0): Transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research*, *28*(1), 65–67. doi:10.1093/nar/28.1.65

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millan-Zarate, D., Diaz-Peredo, E., & Sanchez-Solano, F. (2001). RegulonDB (version 3.2): Transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research*, *29*(1), 72–74. doi:10.1093/nar/29.1.72

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Peralta-Gil, M., Penaloza-Spinola, M. I., & Martinez-Antonio, A. (2006b). The comprehensive updated regulatory network of Escherichia coli K-12. *BMC Bioinformatics*, *7*, 5. doi:10.1186/1471-2105-7-5

Santillan, M., & Mackey, M. C. (2001). Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(4), 1364–1369. doi:10.1073/pnas.98.4.1364

Schadt, E., Lamb, J., Yang, X., Zhu, J., Edwards, S., & Guhathakurta, D. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, *37*(7), 710. doi:10.1038/ng1589

Schäfer, J., & Strimmer, K. (2005a). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics (Oxford, England)*, *21*, 754–764. doi:10.1093/bioinformatics/bti062

Schäfer, J., & Strimmer, K. (2005b). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, *4*, 32. doi:10.2202/1544-6115.1175

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, *31*(1), 64–68. doi:10.1038/ng881

Shimoni, Y., Friedlander, G., Hetzroni, G., Niv, G., Altuvia, S., & Biham, O. (2007). Regulation of gene expression by small non-coding RNAs: A quantitative view. *Molecular Systems Biology*, *3*, 138. doi:10.1038/msb4100181

Spirtes, P., Glymour, C., & Scheines, R. (1993). Causation, prediction, and search. MIT Press.

Steuer, R., Kurths, J., Daub, C. O., Weise, J., & Selbig, J. (2002). The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics (Oxford, England)*, *18*(Suppl 2), S231–S240.

Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(16), 9440–9445. doi:10.1073/pnas.1530509100

Thieffry, D., & Thomas, R. (1998). Qualitative analysis of gene networks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 77–88.

Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, *42*(3), 563–585. doi:10.1016/0022-5193(73)90247-6

Tong, A. H., Lesage, G., Bader, G. D., Ding, H., Xu, H., & Xin, X. (2004). Global mapping of the yeast genetic interaction network. *Science*, *303*(5659), 808–813. doi:10.1126/science.1091317

Tyson, J. J., Chen, K. C., & Novak, B. (2003). Sniffers, buzzers, toggles, and blinkers: Dynamics of regulatory and signaling pathways in the cell. *Current Opinion in Cell Biology*, *15*(2), 221–231. doi:10.1016/S0955-0674(03)00017-6

Veiga, D. F., Vicente, F. F., Grivet, M., de la Fuente, A., & Vasconcelos, A. T. (2007). Genome-wide partial correlation analysis of Escherichia coli microarray data. *Genetics and Molecular Research*, *6*(4), 730–742.

von Bertalanffy, L. (1968). General system theory. New York: Braziller.

Wagner, A. (2001). How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. *Bioinformatics (Oxford, England)*, *17*(12), 1183–1197. doi:10.1093/bioinformatics/17.12.1183

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. doi:10.1038/30918

Weiner, N. (1948). Cybernetics or control and communication in the animal and the machine. Cambridge, MA: MIT Press.

Westerhoff, H. V., & van Dam, K. (1987). Thermodynamics and control of biological free energy transduction. Amsterdam: Elsevier.

Wille, A., & Buhlmann, P. (2006). Low-order conditional independence graphs for inferring genetic networks. Stat. Appl. Genet. Mol. Biol., 5(1), Article 1.

Wille, A., Zimmermann, P., Vranova, E., Furholz, A., Laule, O., & Bleuler, S. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome Biology*, *5*(11), R92. doi:10.1186/gb-2004-5-11-r92

Yeung, M., Tegner, J., & Collins, J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(9), 6163–6168. doi:10.1073/pnas.092576199

Zhu, J., Lum, P. Y., Lamb, J., GuhaThakurta, D., Edwards, S. W., & Thieringer, R. (2004). An integrative genomics approach to the reconstruction of gene networks in segregating populations. *Cytogenetic and Genome Research*, *105*(2-4), 363–374. doi:10.1159/000078209

KEY TERMS AND DEFINITIONS

Co-Expression Network: A network model in which nodes represent gene-activities and the undirected edges represent significant associations.

Cyclic Network: A network with at least one directed path that starts and ends in the same node.

Directed Graph: A network with only directed edges between the nodes.

Gene Regulatory Network: A network model in which nodes represent gene-activities and the directed edges represent direct causal influences and undirected edges represent associations due to confounding.

Hidden Variables: Variables that are not explicietely represented in the network model, often because these have not been experimentally observed.

Mixed Graph: A network with undirected as well as directed edges between the nodes.

Transcriptional Regulatory Network: A network model of transcription factor-target relationships. Directed edges run from transcription factor nodes to target nodes.

Undirected Graph: A network with only undirected edges between the nodes.

Chapter 2
Introduction to GRNs

Ugo Ala
Università di Torino, Italy

Christian Damasco
Università di Torino, Italy

ABSTRACT

The post-genomic era shifted the main biological focus from 'single-gene' to 'genome-wide' approaches. High throughput data available from new technologies allowed to get inside main features of gene expression and its regulation and, at the same time, to discover a more complex level of organization. Analysis of this complexity demonstrated the existence of nonrandom and well-defined structures that determine a network of interactions. In the first part of the chapter, we present a functional introduction to mechanisms involved in genes expression regulation, an overview of network theory, and main technologies developed in recent years to analyze biological processes are discussed. In the second part, we review genes regulatory networks and their importance in system biology.

INTRODUCTION

In the last two decades, biologists have drastically changed their approach to the study of the cell. In the literature, several works describe functional and biochemical analysis focusing on a single gene (Menasche et al., 2003; Miles et al., 2005) or a protein family (Logan et al., 2004; Sasaki et al., 2005). This "single-gene" approach led to a comprehensive knowledge about how or where a single gene of interest works. Recently, some innovative technologies are generating a great amount of biological data and represent a fertile source of knowledge. The most significant of these techniques, described in the *Technology Background* section of this chapter, are DNA microarrays, serial analysis of gene expression (SAGE) and chromatin immunoprecipitation chips (ChIP-chip). The availability of high-throughput data on the role of biological molecules allows a more exhaustive analysis of biological processes, that is the main focus of system biology.

DOI: 10.4018/978-1-60566-685-3.ch002

The need for a tool to integrate high-throughput biological data attracted the attention of the scientific community to the network paradigm as one of the most powerful and versatile theory for the study of complex systems (Albert et al., 2002).

In particular, the network approach offers a theoretical picture that can be used to explain and analyze the structure of biological systems and their evolution. Many theoretical studies on networks have demonstrated their application to model metabolic networks (Fiehn et al., 2003), neuronal networks (Kullander, 2005), gene regulatory networks (GRNs) (Olson, 2006), and other biological networks (Hollenberg, 2007).

What are networks? Networks are simply sets of items, called nodes, joined by specific types of relationships called links.

At the level of gene regulation, the nodes represent genes, proteins, mRNA and biological molecules in general, depending on which molecular products are considered. The links represent molecular interactions such as protein-protein interactions (Vidal et al., 1996), protein-DNA interactions (Gao et al., 2008), gene co-expression (Ala et al., 2008) and others.

Many different kinds of gene networks can be obtained, depending on which particular biological target is considered. Transcriptional regulation is a complex process that involves a great amount of elements and network theory helps to construct a comprehensive view about this process. However, a precise and commonly accepted definition of Gene Regulatory Network (GRN) does not yet exist (Brazma et al., 2003; Dewey et al., 2002). Under this label, it is possible to define various complementary models describing regulatory processes and functional relationships. The most common models are Coexpression Networks (CNs) based on similar expression profiles, Transcription Factors Networks (TFNs) centred on transcription factors activity, Signal Transduction Networks (STNs) that explore gene-activities and causal-effect relationships among genes and proteins under different environmental conditions (as defined in Galperin, 2004; Martelli et al., 2006; Tran et al., 2007) and Genetic Interaction Networks (GINs) that define logical relationships between genes, as defined in (Beyer et al., 2007; Tong et al., 2004), by comparing observed phenotypes of wild-type and mutant individuals of a species. In this chapter, we will focus on CNs and TFNs.

Biological networks can be constructed in different ways: from differential equations (Climescu-Haulica et al., 2007) to statistical correlation integrated by other biological information, such as phylogenetic conservation or gene function (Stuart et al., 2003), to minimize false positives among the inferred interactions, from Bayesian (Mukerjee et al., 2008) to Boolean networks (Martin et al., 2007).

Although the widespread use of experimental data provides an opportunity to investigate GRNs from another point of view, some limitations exist: it is not possible to analyze all genes and evaluate every biological status, information about the variability of expression profiles is lost, and experimental noise decreases data quality.

Some global properties of abstract network models can be used to analyze GRNs: mapping a real network to an abstract model allows the application of statistical inference to detect specific network fetaures. GRNs often display characteristic network features such as short path lengths and high cluster coefficient, typical of highly connected graphs, as described in Barabasi et al. (2004). The degree distribution of a typical GRN is often scale-free and described by a power-law (Albert, 2005), but GRNs could also show small world networks features (Watts et al., 1998). At a smaller scale, GRNs display typical structures as highly connected nodes (hubs), communities and their organization into hierarchical modules (Ravasz et al., 2003).

Applications of GRNs can be classified into two categories: the first one mainly descriptive (qualitative approach) and the second more pragmatic, useful to make predictions (quantitative approach). Qualitative analysis can give an explanation of evolution of the genome and of genetic interactions, thus joining network theory (with particularly regards to the preferential attachment hypothesis) with biological evidence like gene duplications (Bhan et al., 2002; Rzhetsky et al., 2001). On the other hand, quantitative analysis starts from a global point of view to focus again the attention on particular details such as single transcriptional units, functional annotation, relationship to genetic diseases and pathway investigation (Herrgård et al., 2008; Mo et al., 2008).

The starting point of the chapter is a brief exposition of the current knowledge about transcriptional regulation of gene expression with special attention to transcription factors and their interactions. Then, an introduction to network theory is offered, in order to allow the merging of biological information and mathematical model. New high-throughput technologies employed are described in the third section of the Background topic. The main thrust of the chapter is an overview of some pioneering and more recent works on network modelling of biological systems to show how these structures are evolutionarily conserved in Eukaryotes. In the conclusions we suggest possible improvements of GRN analysis and co-operative combination of information focusing on future perspectives in network biology.

BACKGROUND

Biological Background

Analysis of the mechanisms regulating gene expression is one of the most exciting fields of research involving various areas, from molecular and computational biology to molecular genetics, from physics and mathematics to biochemistry. Mechanisms underlying this process became increasingly complex as organisms evolved (Gustincich et al., 2006; Huang et al., 1999; Rockman et al., 2006).

In *Prokaryotes* it is possible to discovery complex and very organized regulation pathways. In the bacterium *Escherichia Coli*, gene expression is strongly regulated by the environment and the availability of source of food. This happens because Bacteria live in environments subjected to frequently changes; to reduce energetic waste, gene transcription in Bacteria is directly regulated by the presence of some metabolites (Madan Babu et al., 2003). To further on optimize regulatory processes, Bacteria show a typical genetic organization called *operon*. The operon is a group of adjacent genes expressed as a single RNA molecule together with their genomic control regions. The expression of an operon is submitted to the presence of the responsible metabolite; its presence (or absence) induces the expression of genes block that it regulates. One of the most studied system is the lactose operon (*lac operon*) and its activation is regulated by the repressor *lac*, the activator protein *CAP* and their interactions with RNA polymerase. Only in the presence of lactose and absence of glucose, maximal transcription of the *lac* operon occurs. In this situation, the lac repressor does not bind to DNA, CAP binds its control region on the DNA and this combination promotes the transcription (Alberts et al., 2002).

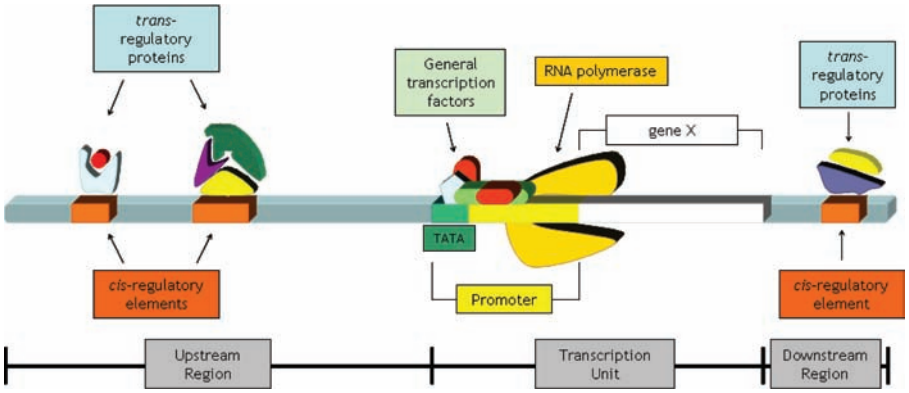
Transcriptional regulation of gene expression in *Eukaryotes* is a crucial step in the definition of the fate of cells and cellular structures (Wray, 2003). The differential expression of genes during developmental stages, cell-cycle phases or across tissues determines the differentiation process of a cell and their future roles. The extensive knowledge about genomes gained in the last years led to the discovery and analysis of the key control mechanisms of gene regulation (Kornberg, 1999) showing a complexity greater than in *Prokaryotes*.

DNA molecules are usually ultra-condensed into structures called *nucleosomes* (Mellor, 2006) where they bound histone proteins to form a complex structure that protect cells from abundant and useless transcription events (a structural state known as *heterochromatin*). To activate transcription, cells need to unclench this structure (Horvath et al., 2001). This first coarse level of regulation is given by molecular modifications that unwrap the condensed structure and release the DNA portions that have to be transcribed (state called *euchromatin*). These changes are essentially due to two molecular processes: the acetylation of histone proteins responsible for chromatin architecture and the methylation of specific regions on DNA strands known as *CpG islands*. This mechanism is controlled by two well-defined classes of enzymes: histone acetyltransferases (HAT) and methyltransferases (HMT). Both acetylation and methylation are very important and they are subject to rigorous patterns that determine cell type-specific gene expression profiles (Fraga et al., 2005; Robertson, 2002). Sometimes, other cellular factors in the cell can bind DNA packaged in a chromatin conformation more accessible and initiate gene transcription by remodelling nucleosomes.

On the unrolled DNA strand, both coding and non-coding sequences become accessible and can interact with factors present in the cell. This configuration allows a control mechanism of transcription regulation based on the binding of proteins of the transcription machinery on specific sequences acting as their substrate (Muller et al. 2004). DNA sequences bound by the machinery are known as *cis*-regulatory elements, genomic sequences different in length mainly located in the non-coding fraction of the double helix. Parallel, *trans*-regulatory elements are DNA binding proteins that regulate transcriptional events interacting with their specific sequence on the genome (Scannell et al, 2004; Wittkopp, 2005). The fundamental *trans*-regulatory element is the enzyme responsible for the effective transcription of DNA, the *RNA-polymerase*. RNA-polymerase binding sites are usually located upstream of and close to the transcription start site (TSS) in the region known as *core promoter*. The polymerase forms the regulatory machinery complex with other very important co-factors that influence its binding to the consensus sequence on the DNA strand.

Many other DNA sequences are binding-sites for eukaryotic gene activators, originally termed *enhancers*, since their presence increases dramatically the rate of transcription acting directly on the polymerase activity. Enhancers are bound by DNA-binding proteins that control gene transcription in a positive (*activators*) or negative (*repressor*) manner. A surprising discovery regarding enhancers was that activator proteins can be bound thousands of nucleotide pairs away from the promoter (Carter et al.,

Figure 1. Diagram of a typical gene control region



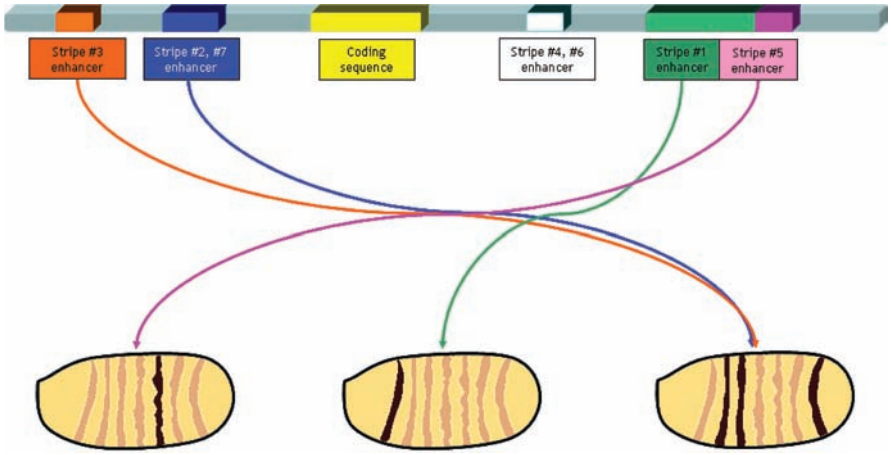
2002). Moreover, they can influence transcription of a gene when bound either upstream or downstream from it or in non-coding regions of a transcription unit. For this reason, today we define a *gene control region* (Figure 1) as the whole expanse of DNA involved in regulating transcription of a gene, including the core promoter, where the general transcription factors (GTFs) and the polymerase assembly, and all of the regulatory sequences to which gene regulatory proteins bind.

Specific regulatory proteins are known as *Transcription Factors* (TFs), proteins that bind well-defined sites on DNA molecules known as DNA-binding domains (specific for each TF) and co-attend to the transcription of genetic information from DNA to mRNA by activating or repressing the process driven by the RNA polymerase (Kadonaga, 2004; Muller, 2001). TFs show a modular structure containing some necessary domains: a DNA-binding domain (DBD) that attaches to specific short DNA sequences and a trans-activating domain (TAD) that contains binding sites for other proteins such as transcriptional coregulators. For example, many TFs are involved in the development of the organism, turning on transcription of genes that regulate cells morphology and differentiation (Wray, 2003). Responses to intracellular signals are often mediated by TFs; cells communicate by releasing molecules producing signalling cascades associated to the upregulation or the downregulation (Brivanlou et al., 2002). Different responses and variations in gene expression are carefully regulated by TFs action. Binding sites for TFs are well-defined; for each TF, in order to determine conserved nucleotides that compose a binding site, we can define a position specific scoring matrix (PSSMs). In a PSSM, for every position, every nucleotide has a score associated to the probability to find it in that position and the global score of the matrix define a consensus sequence (Stormo, 2000). PSSMs are collected into public and commercial databases of matrices like TRANSFAC (Matys et al., 2006) or JASPAR (Bryne et al., 2008). Looking for an over-representation of matrices in gene control regions is a very active research field in computational biology (Brown, 2008; Wasserman et al., 2004). However, evolutionary analysis of genomes and organisms complexity recently showed that the new horizon in studies on transcriptional is represented by combinatorial analysis (van Dijk et al., 2008).

Genome sequencing and analysis of many model organisms confirmed the hypothesis that organisms complexity not only depends on the number of transcription units, but also on the regulatory complexity of their expression (Markstein et al., 2002). The large size of promoter region sequences allows them to host many binding sites for different transcription factors. As a symbolic example, it is sufficient to compare the fruitfly *Drosophila melanogaster* and the nematode *Caenorhabditis elegans*: the fly has less than 14,000 genes, while the worm has about 20,000. However, anatomical, developmental and other biological observations suggest that Drosophila can be considered more complex than *Caenorhabditis elegans*. If the complexity of different species is not directly proportional to the number of genes, what is the element that determines these developmental differences? The answer is hidden behind an elementary concepts widely studied in the last few years: transcription factors act on gene expression regulation not independently, but following a combinatorial and coordinated control mechanism that finely adjusts gene expression profiles for developmental stages, tissues or cell types (Pilpel et al., 2001). Combinatorial and coordinated control means that gene transcription is not regulated by a single signal of activation or repression, but by the correct integration of all signals originating from a combination of transcription factors that are alternatively bound and functionally active.

Considering as an example an extensively studied gene regulation control region like promoter of gene *even-skipped* (*eve*) in *Drosophila Melanogaster* leads to a clear explanation (Janssens et al., 2006). This control region regulates expression of gene eve during different developmental stages and in different positional stripes of larvae. It is composed of 12 partially or totally overlapping binding sites for

Figure 2. Promoter region of the even-skipped gene control specific transcription bands in the Drosophila Melanogaster embryo. Different combination of cis-regulatory modules bound determines different expression patterns.



4 transcription factors that synergistically modulate the transcription rate. Gene expression levels are precisely regulated in different stripes because of different combinations of bound and active transcription factors (figure 2). This illustrates why current research privileges combinatorial analysis of transcription factors and DNA regulatory elements (Morgan et al., 2007).

The mechanisms described so far involve mostly proteins as transcription regulators, but the complexity of multicellular organisms further increases due to another class of regulators.

The RNA-polymerase product is a primary transcript that after the processing driven by cellular enzymes is transformed in the *messenger RNA* (mRNA). At the level of post-transcriptional control, entirely new mechanisms of gene regulations arise; they are mediated by the action of a large class of non-coding RNAs known as microRNAs (miRNAs), which function as repressors in almost all organisms (Ambros, 2004). miRNAs suppress specific transcripts by binding to complementary sequences on the RNA molecules usually located in the untranslated region (3'-UTR) of gene of interest; RNA bound by a miRNA is processed by a couple of enzymes, *Dicer/Drosha*, and degraded inside the cell. Various studies have demonstrated that miRNAs have important roles in animal and plant development (Kloosterman, 2006; Kosik, 2006). Interest in miRNAs and their role in transcriptional regulation has sensationally increased during the last years because only the integration of regulatory signals of both transcription factors and miRNAs can give a comprehensive and unified framework of gene regulation (Chen, 2007).

Post-translational control of gene regulation is the last mechanism to act. After the translation of mRNAs, proteins product can be subjected to modification that increase or reduce their activities or change proteins localization inside the cell. A great number of proteins are substrates of two class of enzymes, called *kinases* and *phosphatases*, that phosphorylate and dephosphorylate them respectively. Many metabolic pathways are regulated through the balanced action of these enzymes (Cohen, 2002).

Other proteins can be engaged with fatty acids chains that translocate them from the cytoplasm to the membrane of the cell. Various protein involved in signal transduction are subjected to these modifications. Two examples are the *Src* family of protein kinases that is myristoylated and the effector protein *Ras* that is anchored to the membrane through a farnesyl-group (Resh, 2006).

A post-translational modification that reduces protein levels in the cell is the ubiqutination. This modification is driven by a group of enzymes that act on the targeted protein functional groups that redirect it to the proteasome, the protein degradation system (Elsasser et al., 2005).

Network Background

Network theory is a field of applied mathematics and physics, deeply related to graph theory. It is applied in a variety of disciplines including sociology, computer science, biology and economics. Network theory concerns the study of graphs as a representation of either symmetric relations (*undirected connections*) or asymmetric relations (*directed connections*) among discrete objects, that can represent human beings in social networks (Wellman, 1998), computers or links in computer science networks (Albert et al., 2002), genes or proteins in biological networks (Barabasi et al., 2004) and enterprises for economic networks (Manski, 2000).

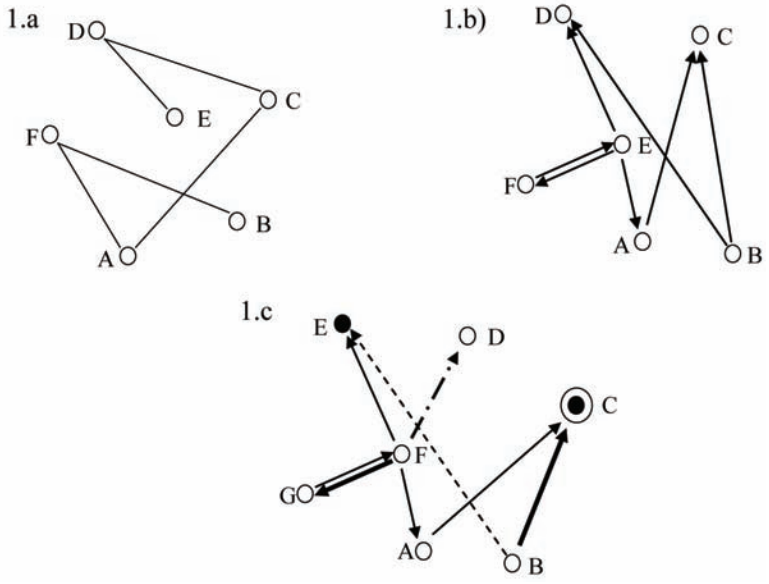
A *network* is simply a set of items, called *nodes* or *vertices*, connected by lines called *links* or *edges*.

A network can be represented by a graph, where links may be undirected, when a line from point A to point B is considered to be the same thing as a line from point B to point A (symmetric relation), or directed, when the two directions are counted as being distinct arcs or directed edges (asymmetric relation).

Such a set of nodes connected by edges represents the simplest kind of network, but we can have different types of vertices (characterized by different information content) and different types of edges (Figure 3).

In a biological context, we consider a type of network called “complex network”, characterized by certain non-trivial topological features that do not occur in simple networks. Such non-trivial features

Figure 3. Some examples of different kinds of networks; 1a) undirect network (edges linking nodes have no directions), 1b) direct network, 1c) direct network with varying node and edge weights.



include: a long tail in the degree distribution, a high clustering coefficient, community structure at many scales and evidence of a hierarchical structure.

We first examine these new concepts that can help in understanding network characteristics and topology (Table 1), then we give a brief definition of two particular kinds of complex networks.

The two most well-known and specific examples of complex networks are *small-world networks* and *scale-free networks*.

A network is called a *small-world network* by analogy with the small-world phenomenon (known as "six degrees of separation"), first tested experimentally by Milgram (1967). The basic result of this experiment was that two arbitrary people are connected on average by approximately six degrees of separation, i.e. the diameter of the corresponding graph of social connections is not much larger than six. The first small-world network model was proposed by Watts and Strogatz (1998). In this model, the transformation of a regular graph, in which the diameter is proportional to the size of the network, into a "small world" one, in which the average number of edges between two vertices is very small (while the clustering coefficient stays large), is obtained and the authors demonstrate that the addition of only a small number of long-range links is required. Summarizing, a graph is considered small-world if:

- the mean shortest distance between nodes pairs scales logarithmically or slower with network size;
- the average clustering coefficient is significantly higher than a random graph constructed on the same vertex set.

A network is named *scale-free* if its degree distribution follows a particular mathematical function called *power law* where few nodes with many links (*hubs*) co-exist among many nodes with few links.

Table 1. Definitions of network characteristics

FEATURE	DEFINITION
Degree	The number of links connected to a node. A direct graph has both an in-degree and an out-degree for each node, corresponding to the number of in-coming and out-going links, respectively.
Degrees distribution	The probability that a node selected at random has a certain number of links.
Clustering coefficient	A measure of the interconnectivity among neighbours of a node *N*. Neighbours of *N* are nodes connected to *N* by an edge.
Average clustering coefficient	The average of the clustering coefficient for each node (Watts et al., 1998). It provides a global measure of how well the neighbors of nodes are locally interconnected.
Community structure	A natural division of the network into sets characterized by groups of nodes that share a high density of internal links and a lower density of links to external nodes (Newman, 2006). In biology, communities are also called modules, motifs or clusters.
Average path length	The average number of steps along the shortest paths for all possible pairs of network nodes (Strogatz, 2001).
Distance	The length in number of edges along the shortest (*geodesic*) path connecting two nodes.
Diameter	The maximal distance between any pair of node of a graph.
Betweenness	The number of shortest paths going through a certain node.
Bottlenecks	Nodes with the highest betweenness. They control most of the information flow in the network, representing critical points of networks (Yu et al, 2007).
Hierarchical organization	In a complex networks implies that small groups of nodes can be organized into increasingly larger groups, maintaining at the same time a scale-free (*see below*) topology (Ravasz et al., 2003).

Networks obtained from lattice models, where every node has roughly the same degree, show a single well-defined scale; in contrast, the *power law* implies that the degree distribution of these networks has no characteristic scale. An example of networks with a well-defined scale is the Erdős–Rényi random graph (Erdős et al., 1959). In a network with a scale-free degree distribution, nodes with a degree that is orders of magnitude larger than the average (*hubs*) are present. The interest in scale-free networks began to flourish in the late '90s with the discovery of a power-law degree distribution in many real world networks such as the World Wide Web, protein interaction networks, and many others. Although many of these distributions are not unambiguously power laws, their particular topology shows that networks characterized by this kind of distribution are very different from what could be expected if edges would be generated at random (for example, by a Poisson distribution). There are many different ways to generate a network with a power-law degree distribution, but the most well known is based on the preferential attachment rule proposed by Barabási and Albert (2002).

The average path length for scale-free networks is smaller than in random graphs, indicating that scale-free topology, more heterogeneous than topology of random graph, deeply affects the distance between nodes; however a theoretical expression giving a good approximation for scale-free model has not been found. Also for the clustering coefficient there is no known analytic model. Observations on some models revealed that the clustering coefficient of scale-free networks decreases with network size following approximately a power law decay: a difference from small-world models, where the cluster coefficient is independent of the size of the network. Networks with a power-law degree distribution can be highly resistant to the random deletion of nodes, since only few hubs are essential for maintaining normal topology: the vast majority of nodes remains connected together in a *giant component* (i.e., a connected sub-graph that contains the majority of the graph's nodes).

Technology Background

A number of array-based technologies has been developed over the last several years, and technological development in this area is likely to continue. These technologies are mainly based on DNA, proteins, antibodies and combinatorial chemistry arrays but every biological molecule could be probably studied with an array-based method. So far, DNA arrays designed to determine gene expression levels in living cells have received the greatest attention. Since they allow simultaneous measurements of thousands of mRNA target molecules and genome probes, they are rapidly producing amounts of raw data on a scale never approached before. We now present an overview of current DNA array technologies and briefly describe also a non-array-based technique to measure gene expression levels based on serial analysis (SAGE) and another innovative approach to study transcriptional regulation based on accessibility of chromatin regions (ChIP-on-Chip).

DNA arrays, also called DNA chips, simultaneously measure the level of mRNAs product in a living cell. A DNA array is defined as an orderly arrangement of tens to hundreds of thousands of unique DNA molecules (called *probes*) of known sequence. Every probe is individually synthesized on a rigid surface or pre-synthesized and then attached to the array platform, dependent on the technology employed. The first method developed is commonly known as *cDNA microarrays* (also called *spotted microarrays*) because probes are usually oligonucleotides, cDNAs or small fragments of polymerase chain reaction (PCR) products that correspond to mRNAs.

Successively, specialized manufacturers optimized the technique and they obtained specific *oligonucleotide microarray* triggering a drastic decrease of cDNA microarrays use. Although oligonucleotide

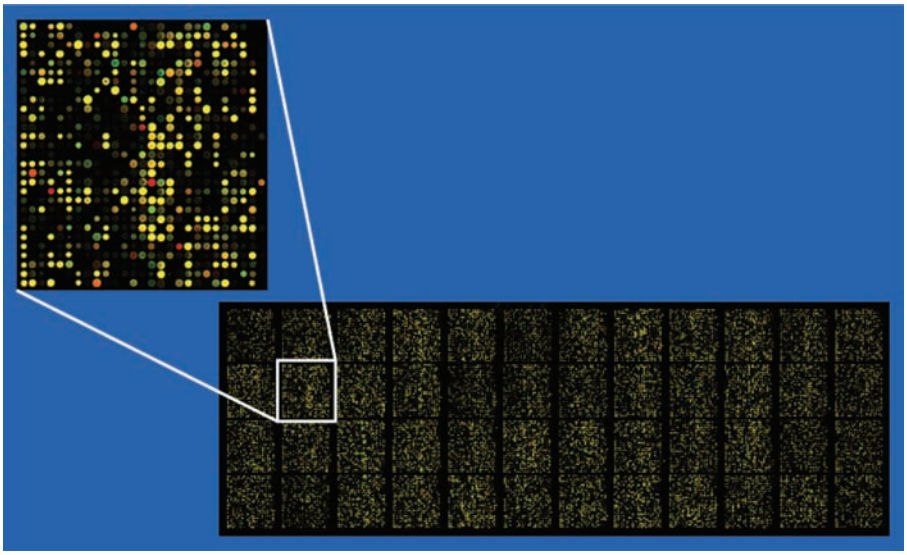
probes are often used in "spotted" microarrays, the term "oligonucleotide microarrays" most often refers to a specific technique of manufacturing. The first synthesis method for manufacturing DNA arrays was the photolithographic method developed by Fodor et al. (1993) and today commercialized by Affymetrix. A set of oligonucleotide probes of 25 nucleotides in length is selected, able to hybridize complementary sequences in target genes of interest. For each gene of interest, all probes matching are collected to define a *probeset*. Statistical software is then used to elaborate raw expression data of probes and to obtain an absolute expression level of a transcript. Other companies, like Agilent, have developed array platforms with a standard piezoelectric (ink-jet) printing process that fix on a glass support longer sequences, up to 60 nucleotides.

Currently, novel approaches to microarrays are rapidly spreading. The most important are *Illumina microarray technology* and *exon-specific arrays*. Illumina company has presented the BeadArray® technology. It yields beads assembled on two substrates, fiber optic bundles or planar silica slides. Each bead is covered with hundreds of thousands of copies of a specific oligonucleotide that act as the capture sequence in an Illumina's assays.

GeneChip® exonarray, instead, is the new technology developed by Affymetrix in which a probeset is associated to each potential exon in the genome. With approximately four probes per exon and roughly 40 probes per gene, exon arrays enable two complementary levels of analysis: simple gene expression and alternative splicing.

When arrays are combined with other techniques and molecules, it becomes possible to obtain new methods to study the transcription. A powerful example is represented by ChIP-on-chip, a technique for the isolation and the identification of the DNA sequences occupied by DNA-binding proteins that combines chromatin immunoprecipitation (ChIP) with microarray technology (chip). The goal of ChIP-on-chip is to locate protein binding sites which results in the identification of functional elements in the genome. The ChIP-on-chip technique was first successfully applied in yeast (Lieb et al., 2001) but today, with little variations in protocols, is also performed on mammalian cells.

Figure 4. Example of an oligonucleotide microarray with enlarged inset to show detail (source: Wikipedia http://en.wikipedia.org/wiki/File:Microarray2.gif)



Finally, gene expression can be evaluated also with another technique called serial analysis of gene expression *(SAGE)* (Velculescu et al., 1995). SAGE analyzes all mRNA molecules in the cell, defined as the *transcriptome*; for each transcript, it is possible to define a small chunk of RNA that unambiguously identifies each RNA molecule. These small pieces, called *tags*, are extracted through the cleavage of restriction enzymes and are linked together in a long chain called *concatemer*. Then, the concatemer is cloned into a vector and long chains produced are sequenced to count the number of small sequence tags for every RNA that compose the chain. This integer number for every transcript is converted in an expression value. The comparison of tags to a specific database determines which ones come from known, well-studied genes and which ones are new.

The diffusion of these technologies allowed the development of specific approaches like time-course and tissue-specific experiments. In time-course experiments cells are dynamically studied during their life cycle and gene expression changes are monitored step-by-step. In this way, for each gene the expression level can be studied as a function of the expression level of all the other genes.

Instead, tissue-specific experiments give more specific information about groups of functionally co-expressed genes and gene expression profiles of tissues and cell lines.

Collapsing information from the transcriptional regulation machinery, the huge amount of data available on biological molecules from the technologies we have described and network theory, it is possible to create models that give a revolutionary way to study genes regulation.

GENE REGULATORY NETWORKS OVERVIEW

The analysis of biological network models produces results (links among genes, communities identification, network topology, etc.) whose mode of analysis can be subdivided into two main categories. As reported in the introduction of the chapter, the first is characterized by qualitative approaches, while the second is focused on quantitative applications.

Qualitative analysis tries to establish a reference frame to explain some aspects of genomes and their evolution. For example, it has been shown that gene regulatory networks grow by duplication (Teichmann et al., 2004). When genes undergo a duplication event, regulatory interactions in networks can be either conserved or lost during the subsequent divergence process (Bhan et al., 2002). Another interesting result is that the development of scale-free networks implies linear preferential attachment (Eriksen et al., 2001). Linear preferential attachment exists when the probability of attachment to a particular node is proportional, at least asymptotically, to the number of links already attached to that node. Combination of these two results, from molecular biology and network theory respectively, opens an intriguing scenario which describes the origin and evolution of highly connected proteins, usually known as *hubs*.

Analysis of network modularity gives another strong contribution to understand how biological networks organization evolves. One of the main contributors to the robustness and evolvability of biological networks is their modularity of function, with modules defined as sets of genes that are strongly interconnected but whose function is separable from those of other modules (Kirschner et al., 1998). Hintze (2008) states that modularity must be a consequence of the evolutionary process, because modularity implies the possibility of change with minimal disruption of function. In particular, the evolution of complex biological networks *in silico* allows to simulate real biological systems to understand their complexity.

Quantitative analysis starts from a global point of view to focus the attention on particular details. Key methods to extract quantitative information from biological networks are the identification of net-

work motifs and communities (Zhang et al., 2007) and the gene clustering that is performed following different algorithms (D'haeseleer, 2005; Zhao et al., 2005). Their application results in the isolation of clusters of genes that can be combined with information stored into various biological databases. For example, the *Gene Ontology* (GO)[1] project offers the necessary information to develop statistical tools looking at the overrepresentation of GO terms within network communities, in order to obtain putative gene functional annotations (Pellegrino et al., 2004). In a similar way, the combination with the *Online Mendelian Inheritance in Man*™ database (OMIM™)[2] gives the possibility to predict disease-related genes (Ala et al., 2008; Lage et al., 2007).

In this section, we review some examples for two of the most studied GRNs. The first part is dedicated to Transcription Factor Networks (TFNs) and to highlight the conservation of network structures from plants to mammals. In the second part, the focus shifts to Coexpression Networks (CNs) where phylogenetic information combined with network theory is used to make functional predictions.

Transcription Factors Networks (TFNs)

TFNs can be inferred directly from experimental results of physical associations between transcription factors and their DNA binding sites defined as PSSMs (Frith et al, 2002). Networks based on transcription factors can be divided in two types.

In the first one, the analysis concerns transcription factors and their target genes. A network built using these elements will show transcriptional factors and genes as nodes and regulatory interactions as edges. In this way, it is possible to highlight cellular signalling pathways. The second type takes into account a smaller version of the previous configuration: only transcription factors are considered, so that they represent nodes linked by a regulatory interaction. Interactions exist when two factors bind the same promoter region of at least one gene, regulating its expression.

In *Arabidopsis thaliana* (thale cress), cell identity during the three main phases of root development (primary root meristem establishment and maintenance, root hair differentiation, and lateral root formation) is controlled by specific transcription factor networks. The analysis of the whole Arabidopsis genome sequence revealed that approximately 5% of the genes encode transcription factors that interact not only with other regulatory proteins but also with the other 95% of the genes (Riechmann et al., 2000). Montiel (2004) states that transcription factors give the opportunity to decrypt gene regulatory networks that control development programs and can be considered as major keys to better understand root tissue differentiation and root development in response to internal growth regulators as well as environmental signals. They also deduce that transcription factors must be considered at a higher level not just for their DNA-binding functions, but rather as crucial members of regulatory networks.

Saccharomyces cerevisiae (yeast) was the first eukaryotic model organism used to study mechanisms of transcriptional regulation. The complexity level of its network is neither trivial nor too high (like that of Mammalian regulatory networks) and the huge amount of expression data available made this unicellular organism the most attractive to test a global scale approach. Yeast studies led to an important new insight: networks of regulator-gene interactions are the background of pathways that are used to regulate global gene expression programs. Extensive studies identified network motifs, the simplest units of network architecture, and demonstrated that these motifs are the building blocks of the transcriptional regulatory process. (Wu et al., 2006).

Rising to a higher complexity level and moving to *Caenhorabditis elegans* (nematode) we can see that network architecture is strongly maintained and very useful to characterize gene regulation. As previ-

ously observed in *Arabidopsis*, also in metazoans 5%–10% of the genes encode predicted transcription factors (Reece-Hoyes et al., 2005), each of which regulates the expression of one or more target genes. Vermeirssen (2007) evaluates protein-DNA interactions between transcription factors and their target genes. He shows, for example, that the core neuronal protein-DNA interactions network is organized into two transcription factors modules. Moreover, this study represents an important step because for the first time the subdivision into clusters of a metazoan protein–DNA interactions network defines function-specific transcription factors modules.

Studies on *Drosophila melanogaster* (fruitfly) confirm the effectiveness of representing transcription factors interactions as a network (Aerts et al., 2007; Fowlkes et al., 2007; Segal et al., 2007). The segmentation genes network is a common example to explain the role of transcription control in pattern formation (Scott et al., 1987). The regulation within this network is almost entirely transcriptional, and *cis*- and *trans*-acting components are well characterized. The network includes maternal and zygotic factors that act in a four-tiered hierarchical fashion to generate increasingly refined and complex expression patterns along the anterior-posterior axis in the blastoderm embryo (Schroeder et al., 2004). In this case, the global analysis allows to show strongly connected modules and signalling activation or repression cascades that traditional single-gene approaches cannot easily unravel.

Mammals, and in particular *Mus musculus* (mouse) and *Homo sapiens*, represent the most difficult test-bed and at the same time the main goal of the application of network theory to transcription factor analysis and modelling. A large number of studies were published in the last few years (Duncan et al., 1998; Zenke et al., 2006) and some mathematical and statistical methods were developed (Rastegar et al., 2000). From the biological point of view, advances in transgenic mice production made possible to obtain specific experimental data. Exploiting these innovations, Maroulakou (2000) demonstrates the need of a network of Ets transcription factors family to maintain tissue remodelling and integrity, in particular during embryonic developmental stages in mammals. The large family of Ets transcription factors control a spectrum of developmental processes and nearly 30 mammalian family members have been isolated (Dejana et al., 2007). Actions of Ets transcription factors expressed at different levels are crucial for hematopoietic and endothelial cells development. The authors conclude that to investigate the roles of the Ets family of transcription factors, mammalian models based on a network of Ets genes and their targets, rather than on a single gene in a pathway, are necessary, as we argued previously. These results, first shown for the Ets family were successively found in all transcription factors families (Kang et al., 2005; Tsantoulis et al., 2005). Moreover, more recent approaches to TFNs integrate both computational and molecular biology techniques. As described by Kel (2004), one suitable approach is to develop genetic algorithms to analyze global gene expression microarrays. Their computational strategy analyzes the promoters of genes regulated by aryl hydrocarbon receptor (AhR) with a genetic algorithm previously described by Kel-Margoulis (2002). The analysis reveals a network of transcription factors with several feedback loops and signalling cascades. This network of transcription factors can also explain the regulation of several genes that are not direct targets of AhR binding. Their regulation can be mediated through other transcription factors directly regulated by AhR.

Coexpression Networks (CNs)

Coexpression Networks (CNs) can be inferred from microarrays experiments, a very powerful technology that allows to simultaneously measure the expression level of thousands of genes as described in the “Technology background” section. Microarray data are stored in matrices where rows (*i*) are related

Table 2. An example of a matrix from microarray experiments containing n rows and m columns

	Exp 1	**Exp 2**	**…**	**Exp** *j*	**…**	**Exp** *m*
Probe 1	Value (1,1)	Value (1,2)	…	Value (1,j)	…	Value (1,m)
Probe 2	Value (2,1)	Value (2,2)	…	Value (2,j)	…	Value (2,m)
Probe 3	Value (3,1)	Value (3,2)	…	Value (3,j)	…	Value (3,m)
…	…	…	…	…	…	…
…	…	…	…	…	…	…
Probe *i*	Value (i,1)	Value (i,2)	…	Value (i,j)	…	Value (i,m)
…	…	…	…	…	…	…
…	…	…	…	…	…	…
Probe *n*	Value (n,1)	Value (n,2)	…	Value (n,j)	…	Value (n,m)

to probes, representing genes, and columns (*j*) are different experimental conditions; for each matrix element (*i*,*j*) an expression value is reported (table 2).

In a CN, the abstraction from biological data to a mathematical model is realized by mapping genes to nodes and putting edges representing similarity of gene expression according to a given quantitative notion of similarity (or dissimilarity). Given two genes in an expression matrix it is possible to use different quantitative measures of coexpression to construct different coexpression networks. Here, we present two of the most often used dissimilarity measures to evaluate coexpression. A coexpression link exists when the dissimilarity measure between two genes is lower than a defined cutoff. Let *X* and *Y* be two genes and their expression values for the N columns of the matrix: the expression data are real numbers for microarray data and integer counts for SAGE.

The Pearson linear dissimilarity is defined as:

$$dP\left(X,Y\right)=\frac{1-r\left(X,Y\right)}{2}$$

where *r* is the Pearson correlation coefficient defined as:

$$r=\frac{\sum XY-\frac{\sum X\sum Y}{N}}{\sqrt{\left(\sum X^2-\frac{\left(\sum X\right)^2}{N}\right)\left(\sum Y^2-\frac{\left(\sum Y\right)^2}{N}\right)}}$$

The Euclidean distance is:

$$dE\left(X,Y\right)=\sqrt{\sum_{i=1}^{N}\left(x_i-y_i\right)^2}$$

These two measures can be applied to both microarray and SAGE data; in addition, measures specifically targeted to SAGE data and based on Poisson distribution were also developed (Cai et al., 2004).

Data from high-throughput gene expression measurements are affected by a relatively high level of noise; it is therefore necessary to adopt specific strategies to reach a good compromise between specificity and sensitivity of the statistical analysis. Two most common approaches to prevent these problems are:

- the imposition of a more stringent cutoff for dissimilarity measures;
- the use of filters to select interactions between coexpressed genes that share also other biological features.

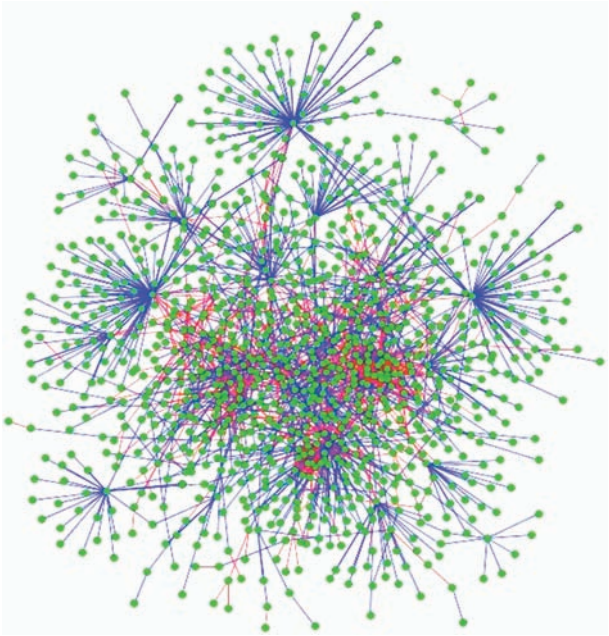
A pioneer work in the landscape of CNs was performed by Stuart (2003). In order to elucidate gene function on a global scale, they identified pairs of genes that are coexpressed over DNA microarrays from multiple species. The filter employed in this work was the phylogenetic conservation because the coexpression of orthologous gene pairs confers a selective advantage and therefore indicates a functional relationship. Four species were compared in the phylogenetic analysis: *Saccharomyces Cerevisiae*, *Caenhorabditis Elegans*, *Drosophila Melanogaster*, *Homo Sapiens*. The use of species not so close from the evolutionary point of view increased the efficiency and selectivity of the filter but allows to study only genes involved in core biological functions. They found that the distribution of gene expression links in the gene-coexpression network is highly non-random, containing significantly more nodes with a larger number of gene expression links than random networks obtained from the same microarray data after permutation. The connectivity of the network followed a power-law distribution suggesting the existence of a selective force in the overall design of genetic pathways to maintain a highly connected class of genes. Finally, the predictions referred to proliferation function for several genes implied by some of these links have been experimentally confirmed.

Later, Jordan (2004) reduced the number of species focusing only on human and mouse in order to increase the knowledge about mammalian gene regulation. In addition, instead of heterogeneous experiments, he took into account only data coming from tissue-specific datasets (Su et al., 2002). The similarity measure choosen in the study is the Pearson correlation coefficient (r). The cutoff for the correlation was set to $r > 0.7$ since for this value the distribution showed a good fit to a generalized Pareto distribution with a power law tail which implied asymptotically scale-free properties and, at the same time, retained enough data for significant statistical analysis. In this case, a small number of hubs emerged and characterized this kind of scale-free networks.

The approach of Lee (2004) was more selective because it extracted homo-specific relationships among genes from multiple human microarray datasets. The coexpression analysis was based on the standard Pearson correlation coefficient and performed independently on the collected datasets. Two genes were defined as “coexpressed” if a statistical significant coexpression was observed in more than one dataset. In addition, anticorrelation was examined and the comparison with correlation showed that the latter is much more significant. A possible explanation proposed for this result is that biological meaningful of negative correlations are harder to detect using microarrays.

Finally, we summarize the work of Ala (2008). The goal was the generation of human-mouse conserved coexpression networks, in order to develop a predictor for unknown gene-disease relationships based on OMIM catalogue. In this case, experimental data were collected from various tissues (Roth et al., 2006; Su et al., 2004) and cell lines (Sherlock et al., 2001) and they were used to generate two human-mouse conserved coexpression networks (CCNs), based on Affymetrix and cDNA microarray platforms, respectively.

Figure 5. A graph of a conserved gene coexpression network from Ala et al. (2008). Links highlight coexpressed genes among different experimental conditions.



Both networks contained a large connected component with some other small connected components containing only a few nodes. As expected from previous studies on gene coexpression networks (Stuart et al., 2003; van Driel et al., 2006), the two networks were topologically similar to other biological networks, characterized by the existence of a few hubs, but they showed a connectivity distribution more similar to an exponential law than to a power law.

Despite previously described works led to many important and original results, a more holistic view on GRNs could include "younger" transcriptional regulation factors like miRNAs (Ke et al., 2003). Actually, a "miRNAs-only" network does not exist but it is known that miRNAs activity must be considered as an integral part of the complex regulation network. Tsang (2007) give a demonstration of this new paradigm observing that miRNA-containing networks have recurrent circuit motifs (usually defined *feedback* and *feedforward loops*) corresponding to positive and negative transcriptional coregulation of a miRNA and its targets. Using gene expression data analyzed with a specific computational pipeline, they show the existence in mammals of two classes of circuits, corresponding to positive and negative transcriptional coregulation of a miRNA and its targets.

Problems

Network theory applied to biology has drastically changed biological research and has offered very powerful instruments to tackle unsolved problems. However, these instruments still have important limits: intrinsic limits in biological techniques to detect the level and the activity of biological molecules inside the cell and the optimization of mathematical models.

Although great advances were made in the last years, microarrays and gene expression measure techniques are still affected by some problems already highlighted in theoretical work as reported by Chu (2003).

A relatively novel technology as oligonucleotide microarray is influenced by systematic errors (Eads et al., 2006). For example, probesets fixed on the support are not always correctly mapping over target-genes sequences; this manufacturer error defines incorrect probesets annotations and, successively, a wrong association of expression values to genes. Technical mistakes in printing or preparation and labelling of samples can generate problems with microarray hybridizations that range from no signal detection to data of apparently high quality that nevertheless are artefacts. After the hybridization, scanner-software read microarray output images. They are based on the elaboration of pixel intensities (or colours, depending on technology employed) to obtain for each sample a correspondent numerical value. After that, raw data obtained from software are normalized with statistical analysis and at the current time, there is not a unique normalization algorithm. The most common are 'Significance Analysis of Microarray' (*SAM,*Tusher et al., 2001), 'Microarray Affymetrix Suite' (*MAS,*Hubbell et al., 2002) and 'Robust Multichip Average' (*RMA,*Irizarry et al., 2003; Katz et al., 2006) algorithms. As shown in Lim's work (2007), depending on what we want to learn from microarray data, the choice of the algorithm is fundamental.

Gene expression is also affected by stochastic regulatory events that occur when transcriptional regulators are present at very low concentrations, so that binding and release of regulators from their binding sites become stochastic events. In these conditions, current high-throughput technologies are not able to correctly quantify very low proteins levels. The suspicion that stochasticity had a significant effect on genes expression came from the observation that genetically identical cells diverge phenotypically.

The work of Elowitz (2002), based on a single-cell approach, enabled determination of two mechanisms by which stochasticity (or noise) is generated. The first one is the *extrinsic noise* generated from fluctuations in the amount, activity or location of cellular components, such as transcription factors or RNA polymerase, that regulate genes transcription. These fluctuations depend on temporal or spatial variation that determine a probability that a gene will be activated or not.

Instead, *intrinsic noise* is linked to random microscopic events that govern reactions occurred in genes transcription. Intrinsic noise is a very subtle snag because, also in a hypothetical cell population where cellular components are expressed at the same concentrations, it is responsible for variation in the expression rate among cells.

CONCLUSIONS AND PERSPECTIVES

Transcriptional regulation is a key process in the life cycle of a cell and many biological molecules that contribute to control it are well-known (Rockman et al., 2006). During evolution, transcriptional regulation significantly changed and its complexity increased as demonstrated by the much more complex regulation of higher eukaryotic genes than prokaryotic ones (Adami et al. 2000).

The approaches described in this chapter showed the existence and evolutionary conservation among many species of GRNs, demonstrating their fundamental role for living organisms.

Recently, network theory was successfully combined with transcriptional regulation and other biological processes allowing to handle the complexity of cellular systems, even if the technologies employed can be improved and mathematical modelling can be optimized. In order to obtain more precise and correct information from experiments, the basic feature is the optimization of experimental design, data acquisition and analysis. Successively, the data produced need appropriate statistical and dynamical model to be integrated together. The current models are promising, but do not take into account all the factors involved in biological processes.

A comprehensive combination of biological data and mathematical models originating in different contexts opened the way for the rapid progress of system biology.

For example, the meeting of genomics and pharmacology is resulted in the origin of pharmacogenomics that studies what target molecules inside the cell are bound by the therapeutic molecules tested. Current pharmacogenomics research is focused on drug discovery, that is the scan of peptides library to search interactions between peptides tested and host target molecules. A new paradigm for drug target selection takes into account global network regulatory interactions among molecules in the genome.

Another example is the comparison between GRNs extracted from data available on gene expression in normal and cancer-affected tissues. This comparison brings to light genes that are differentially expressed in tumors as compared to normal tissue, determining gene-signatures for different tumours. These genes selections combined with the analysis of their regulatory sequences could be employed as diagnostic markers to predict the cancer predisposition in patients and responsible elements in the genome.

Looking at the amount of biological databases available, a great challenge of system biology is the integration of information coming from many biological fields of research (like genome sequencing, gene expression, protein domains, protein-protein interactions, etc.) and constantly increasing.

Now, like never before, network biology and GRNs analysis are employed in many original applications but the integration of well-defined functional biological maps (genomes, proteomes, transcriptomes, phenome, etc.) into an exhaustive model is necessary. Future research will go in this direction and will focus on the optimization of methods and new applications.

REFERENCES

Adami, C., Ofria, C., & Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(9), 4463–4468. doi:10.1073/pnas.97.9.4463

Aerts, S., van Helden, J., Sand, O., & Hassan, B. A. (2007). Fine-tuning enhancer models to predict transcriptional targets across multiple genomes. *PLoS ONE*, *2*(11), e1115. doi:10.1371/journal.pone.0001115

Ala, U., Piro, R. M., Grassi, E., Damasco, C., Silengo, L., & Oti, M. (2008). Prediction of human disease genes by human-mouse conserved coexpression analysis. *PLoS Computational Biology*, *4*(3), e1000043. doi:10.1371/journal.pcbi.1000043

Albert, R. (2005). Scale-free networks in cell biology. *Journal of Cell Science*, *118*, 4947–4957. doi:10.1242/jcs.02714

Albert, R., & Barabasi, A. L. (2002). Statistical mechanic of complex networks. *Reviews of Modern Physics*, *74*(47).

Ambros, V. (2004). The function of animal microRNAs. *Nature*, *431*, 350–355. doi:10.1038/nature02871

Barabási, A. L., & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews. Genetics*, *5*(2), 101–113. doi:10.1038/nrg1272

Beyer, A., Bandyopadhyay, S., & Ideker, T. (2007). Integrating physical and genetic maps: From genomes to interaction networks. *Nature Reviews. Genetics*, *8*, 699–710. doi:10.1038/nrg2144

Bhan, A., Galas, D. J., & Dewey, T. G. (2002). A duplication growth model of gene expression networks. *Bioinformatics (Oxford, England)*, *18*(11), 1486–1493. doi:10.1093/bioinformatics/18.11.1486

Brazma, A., & Schlitt, T. (2003). Reverse engineering of gene regulatory networks: A finite state linear model. *Genome Biology*, *4*(6). doi:10.1186/gb-2003-4-6-p5

Brivanlou, A. H., & Darnell, J. E. Jr. (2002). Signal transduction and the control of gene expression. *Science*, *295*(5556), 813–818. doi:10.1126/science.1066355

Brown, C. T. (2008). Computational approaches to finding and analyzing cis-regulatory elements. *Methods in Cell Biology*, *87*, 337–365. doi:10.1016/S0091-679X(08)00218-5

Byrne, J. C., Valen, E., Tang, M. H., Marstrand, T., Winther, O., & da Piedade, I. (2008). JASPAR, the open access database of transcription factor-binding profiles: New content and tools in the 2008 update. *Nucleic Acids Research*, *36*(Database issue), D102–D106. doi:10.1093/nar/gkm955

Cai, L., Huang, H., Blackshaw, S., Liu, J. S., Cepko, C., & Wong, W. H. (2004). Clustering analysis of SAGE data using a Poisson approach. *Genome Biology*, *5*(7), R51. doi:10.1186/gb-2004-5-7-r51

Carter, D., Chakalova, L., Osborne, C. S., Dai, Y. F., & Fraser, P. (2006). Long-range chromatin regulatory interactions in vivo. *Nature Genetics*, *32*(4), 623–626. doi:10.1038/ng1051

Chen, K., & Rajewsky, N. (2007). The evolution of gene regulation by transcription factors and microRNAs. *Nature Reviews. Genetics*, *8*(2), 93–103. doi:10.1038/nrg1990

Chu, T., Glymour, C., Scheines, R., & Spirtes, P. (2003). A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays. *Bioinformatics (Oxford, England)*, *19*(9), 1147–1152. doi:10.1093/bioinformatics/btg011

Climescu-Haulica, A., & Quirk, M. D. (2007). A stochastic differential equation model for transcriptional regulatory networks. *BMC Bioinformatics*, *8*(Suppl 5), S4. doi:10.1186/1471-2105-8-S5-S4

Cohen, P. (2002). The origins of protein phosphorylation. *Nature Cell Biology*, *4*(5), 127–130. doi:10.1038/ncb0502-e127

D'haeseleer, P. (2005). How does gene expression clustering work? *Nature Biotechnology*, *23*(12), 1499–1501. doi:10.1038/nbt1205-1499

Dejana, E., Taddei, A., & Randi, A. M. (2007). Foxs and Ets in the transcriptional regulation of endothelial cell differentiation and angiogenesis. *Biochimica et Biophysica Acta*, *1775*(2), 298–312.

Dewey, G. T., & Galas, D. J. (2002). In *Eurekah bioscience collection. Landes biosciences*.

Duncan, S. A., Navas, M. A., Dufort, D., Rossant, J., & Stoffel, M. (1998). Regulation of a transcription factor network required for differentiation and metabolism. *Science*, *281*(5377), 692–695. doi:10.1126/science.281.5377.692

Eads, B., Cash, A., Bogart, K., Costello, J., & Andrews, J. (2006). Troubleshooting microarray hybridizations. *Methods in Enzymology*, *411*, 34–39. doi:10.1016/S0076-6879(06)11003-4

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*(5584), 1183–1186. doi:10.1126/science.1070919

Elsasser, S., & Finley, D. (2005). Delivery of ubiquitinated substrates to protein-unfolding machines. *Nature Cell Biology*, *7*(8), 742–749. doi:10.1038/ncb0805-742

Erdős, P., & Rényi, A. (1959). On random graphs. *Publicationes Mathematicae*, *6*, 290–297.

Eriksen, K. A., & Hornquist, M. (2001). Scale-free growing networks imply linear preferential attachment. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *65*(1 pt 2).

Fiehn, O., & Weckwerth, W. (2003). Deciphering metabolic networks. *European Journal of Biochemistry*, *270*(4), 579–588. doi:10.1046/j.1432-1033.2003.03427.x

Fodor, S. P., Rava, R. P., Huang, X. C., Pease, A. C., Holmes, C. P., & Adams, C. L. (1993). Multiplexed biochemical assays with biological chips. *Nature*, *364*, 555–556. doi:10.1038/364555a0

Fowlkes, C. C., Hendriks, C. L., Keränen, S. V., Weber, G. H., Rübel, O., & Huang, M. Y. (2007). A quantitative spatiotemporal atlas of gene expression in the Drosophila blastoderm. *Cell*, *133*(2), 364–374. doi:10.1016/j.cell.2008.01.053

Fraga, M. F., & Esteller, M. (2005). Towards the human cancer epigenome: A first draft of histone modifications. *Cell Cycle (Georgetown, Tex.)*, *4*(10), 1377–1381.

Frith, M. C., Spouge, J. L., Hansen, U., & Weng, Z. (2002). Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Research*, *30*(14), 3214–3224. doi:10.1093/nar/gkf438

Galperin, M. Y. (2004). Bacterial signal transduction network in a genomic perspective. *Environmental Microbiology*, *6*(6), 552–567. doi:10.1111/j.1462-2920.2004.00633.x

Gao, J., Li, W. X., Feng, S. Q., Yuan, Y. S., Wan, D. F., Han, W., & Yu, Y. (2008). (in press). A protein-protein interaction network of transcription factors acting during liver cell proliferation. *Genomics*.

Gustincich, S., Sandelin, A., Plessy, C., Katayama, S., Simone, R., & Lazarevic, D. (2006). The complexity of the mammalian transcriptome. *The Journal of Physiology*, *575*(2), 321–332. doi:10.1113/jphysiol.2006.115568

Herrgård, M. J., Swainston, N., Dobson, P., Dunn, W. B., Arga, K. Y., & Arvas, M. (2008). A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nature Biotechnology*, *26*(10), 1155–1160. doi:10.1038/nbt1492

Hintze, A., & Adami, C. (2008). Evolution of complex modular biological networks. *PLoS Computational Biology*, *4*(2), e23. doi:10.1371/journal.pcbi.0040023

Hollenberg, D. (2007). On the evolution and dynamics of biological networks. *Rivista di Biologia*, *100*(1), 93–118.

Horvath, J. E., Bailey, J. A., Locke, D. P., & Eichler, E. E. (2001). Lessons from the human genome: Transitions between euchromatin and heterochromatin. *Human Molecular Genetics*, *10*(20), 2215–2223. doi:10.1093/hmg/10.20.2215

Huang, L., Guan, R. J., & Pardee, A. B. (1999). Evolution of transcriptional control from prokaryotic beginnings to eukaryotic complexities. *Critical Reviews in Eukaryotic Gene Expression*, *9*(3-4), 175–182.

Hubbell, E., Liu, W. M., & Mei, R. (2002). Robust estimators for expression analysis. *Bioinformatics (Oxford, England)*, *18*(12), 1585–1592. doi:10.1093/bioinformatics/18.12.1585

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., & Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics (Oxford, England)*, *4*(2), 249–264. doi:10.1093/biostatistics/4.2.249

Janssens, H., Hou, S., Jaeger, J., Kim, A. R., Myasnikova, E., Sharp, D., & Reinitz, J. (2006). Quantitative and predictive model of transcriptional control of the Drosophila melanogaster even skipped gene. *Nature Genetics*, *38*(10), 1159–1165. doi:10.1038/ng1886

Jordan, I. K., Marino-Ramirez, L., Wolf, Y. I., & Koonin, E. V. (2004). Conservation and coevolution in the scale-free human gene coexpression network. *Molecular Biology and Evolution*, *21*(11), 2058–2070. doi:10.1093/molbev/msh222

Kadonaga, J. T. (2004). Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors. *Cell*, *116*(2), 247–257. doi:10.1016/S0092-8674(03)01078-X

Kang, H. C., Chae, J. H., Lee, Y. H., Park, M. A., Shin, J. H., & Kim, S. H. (2005). Erythroid cell-specific alpha-globin gene regulation by the CP2 transcription factor family. *Molecular and Cellular Biology*, *25*(14), 6005–6020. doi:10.1128/MCB.25.14.6005-6020.2005

Katz, S., Irizarry, R. A., Lin, X., Tripputi, M., & Porter, M. W. (2006). A summarization approach for Affymetrix GeneChip data using a reference training set from a large, biologically diverse database. *BMC Bioinformatics*, *7*, 464. doi:10.1186/1471-2105-7-464

Ke, X. S., Liu, C. M., Liu, D. P., & Liang, C. C. (2003). MicroRNAs: Key participants in gene regulatory networks. *Current Opinion in Chemical Biology*, *7*(4), 516–523. doi:10.1016/S1367-5931(03)00075-9

Kel, A., Reymann, S., Matys, V., Nettesheim, P., Wingender, E., & Borlak, J. (2004). A novel computational approach for the prediction of networked transcription factors of aryl hydrocarbon-receptor-regulated genes. *Molecular Pharmacology*, *66*(6), 1557–1572. doi:10.1124/mol.104.001677

Kel-Margoulis, O. V., Ivanova, T. G., Wingender, E., & Kel, A. E. (2002). Automatic annotation of genomic regulatory sequences by searching for composite clusters. *Pacific Symposium on Biocomputing*, 187-198.

Kirschner, M., & Gerhart, J. (1998). Evolvability. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(15), 8420–8427. doi:10.1073/pnas.95.15.8420

Klingler, M., Soong, J., Butler, B., & Gergen, J. P. (1996). Disperse versus compact elements for the regulation of runt stripes in *Drosophila*. *Developmental Biology*, *177*(1), 73–84. doi:10.1006/dbio.1996.0146

Kloosterman, W. P., & Plasterk, R. H. (2006). The diverse functions of microRNAs in animal development and disease. *Developmental Cell*, *11*(4), 441–450. doi:10.1016/j.devcel.2006.09.009

Kornberg, R. D. (1999). Eukaryotic transcriptional control. *Trends in Cell Biology*, *9*(12), M46–M49. doi:10.1016/S0962-8924(99)01679-7

Kosik, K. S. (2006). The neuronal microRNA system. *Nature Reviews. Neuroscience*, *7*(12), 911–920. doi:10.1038/nrn2037

Kullander, K. (2005). Genetics moving to neuronal networks. *Trends in Neurosciences*, *28*(5), 239–247. doi:10.1016/j.tins.2005.03.001

Lage, K., Karlberg, E. O., Størling, Z. M., Olason, P. I., Pedersen, A. G., & Rigina, O. (2007). A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature Biotechnology*, *25*(3), 309–316. doi:10.1038/nbt1295

Lee, H. K., Hsu, A. K., Sajdak, J., Qin, J., & Pavlidis, P. (2004). Coexpression analysis of human genes across many microarray data sets. *Genome Research*, *14*(6), 1085–1094. doi:10.1101/gr.1910904

Lieb, J. D., Liu, X., Botstein, D., & Brown, P. O. (2001). Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association. *Nature Genetics*, *28*, 327–334. doi:10.1038/ng569

Logan, N., Delavaine, L., Graham, A., Reilly, C., Wilson, J., & Brummelkamp, T. R. (2004). E2F-7: A distinctive E2F family member with an unusual organization of DNA-binding domains. *Oncogene*, *23*(30), 5138–5150. doi:10.1038/sj.onc.1207649

Madan Babu, M., & Teichmann, S. A. (2003). Functional determinants of transcription factors in Escherichia Coli: Protein families and binding sites. *Trends in Genetics*, *19*(2), 75–79. doi:10.1016/S0168-9525(02)00039-2

Markstein, M., Markstein, P., Markstein, V., & Levine, M. (2002). Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the Drosophila embryo. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(2), 763–768. doi:10.1073/pnas.012591199

Maroulakou, I. G., & Bowe, D. B. (2000). Expression and function of Ets transcription factors in mammalian development: A regulatory network. *Oncogene*, *19*(55), 6432–6442. doi:10.1038/sj.onc.1204039

Martelli, A. M., Nyåkern, M., Tabellini, G., Bortul, R., Tazzari, P. L., Evangelisti, C., & Cocco, L. (2006). Phosphoinositide 3-kinase/Akt signaling pathway and its therapeutical implications for human acute myeloid leukemia. *Leukemia*, *20*(6), 911–928. doi:10.1038/sj.leu.2404245

Martin, S., Zhang, Z., Martino, A., & Faulon, J. L. (2007). Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics (Oxford, England)*, *23*(7), 866–874. doi:10.1093/bioinformatics/btm021

Matys, V., Kel-Margoulis, O. V., Fricke, E., Liebich, I., Land, S., & Barre-Dirrie, A. (2006). TRANSFAC and its module TRANSCompel: Transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, *34*(Database issue), D108–D110. doi:10.1093/nar/gkj143

Mellor, J. (2006). Dynamic nucleosomes and gene transcription. *Trends in Genetics*, *22*(6), 320–329. doi:10.1016/j.tig.2006.03.008

Menasche, G., Feldmann, J., Houdusse, A., Desaymard, C., Fischer, A., Goud, B., & de Saint Basile, G. (2003). Biochemical and functional characterization of Rab27a mutations occurring in Griscelli syndrome patients. *Blood*, *101*(7), 2736–2742. doi:10.1182/blood-2002-09-2789

Miles, M. C., Janket, M. L., Wheeler, E. D., Chattopadhyay, A., Majumder, B., & Dericco, J. (2005). Molecular and functional characterization of a novel splice variant of ANKHD1 that lacks the KH domain and its role in cell survival and apoptosis. *The FEBS Journal*, *27*(16), 4091–4102. doi:10.1111/j.1742-4658.2005.04821.x

Milgram, S. (1967, May). The small world problem. *Psychology Today*, 60–67.

Mo, M. L., & Palsson, B. O. (2008). Understanding human metabolic physiology: a genome-to-systems approach. *Trends Biotechnology.* Retrieved from http://dx.doi.org/10.1016/j.tibtech.2008.09.007

Montiel, G., Gantet, P., Jay-Allemand, C., & Breton, C. (2004). Transcription factor networks. Pathways to the knowledge of root development. *Plant Physiology*, *136*(3), 3478–3485. doi:10.1104/pp.104.051029

Morgan, X. C., Ni, S., Miranker, D. P., & Iyer, V. R. (2007). Predicting combinatorial binding of transcription factors to regulatory elements in the human genome by association rule mining. *BMC Bioinformatics*, *8*, 445. doi:10.1186/1471-2105-8-445

Mukherjee, S., & Speed, T. P. (2008). Network inference using informative priors. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(38), 14313–14318. doi:10.1073/pnas.0802272105

Muller, C. W. (2001). Transcription factors: Global and detailed views. *Current Opinion in Structural Biology*, *11*(1), 26–32. doi:10.1016/S0959-440X(00)00163-9

Muller, F., & Tora, L. (2004). The multicoloured world of promoter recognition complexes. *The EMBO Journal*, *23*(1), 2–8. doi:10.1038/sj.emboj.7600027

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(23), 8577–8582. doi:10.1073/pnas.0601602103

Olson, E. N. (2006). Gene regulatory networks in the evolution and development of the heart. *Science*, *313*(5795), 1922–1927. doi:10.1126/science.1132292

Pellegrino, M., Provero, P., Silengo, L., & Di Cunto, F. (2004). CLOE: Identification of putative functional relationships among genes by comparison of expression profiles between two species. *BMC Bioinformatics*, *5*, 179. doi:10.1186/1471-2105-5-179

Pilpel, Y., Sudarsanam, P., & Church, G. M. (2001). Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, *29*(2), 153–159. doi:10.1038/ng724

Rastegar, M., Lemaigre, F. P., & Rousseau, G. G. (2000). Control of gene expression by growth hormone in liver: Key role of a network of transcription factors. *Molecular and Cellular Endocrinology*, *164*(1-2), 1–4. doi:10.1016/S0303-7207(00)00263-X

Ravasz, E., & Barabási, A. L. (2003). Hierarchical organization in complex networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *67*(2). doi:10.1103/PhysRevE.67.026112

Reece-Hoyes, J. S., Deplancke, B., Shingles, J., Grove, C. A., Hope, I. A., & Walhout, A. J. (2005). A compendium of *Caenorhabditis elegans* regulatory transcription factors: A resource for mapping transcription regulatory networks. *Genome Biology*, *6*(13), R110. doi:10.1186/gb-2005-6-13-r110

Resh, M. D. (2006). Trafficking and signaling by fatty-acylated and prenylated proteins. *Nature Chemical Biology*, *2*(11), 584–590. doi:10.1038/nchembio834

Riechmann, J. L., Heard, J., Martin, G., Reuber, L., Jiang, C., & Keddie, J. (2000). *Arabidopsis* transcription factors: Genomewide comparative analysis among eukaryotes. *Science*, *290*(5499), 2105–2110. doi:10.1126/science.290.5499.2105

Robertson, K. D. (2002). DNA methylation and chromatin–unravelling the tangled web. *Oncogene*, *21*(35), 5361–5379. doi:10.1038/sj.onc.1205609

Rockman, M. V., & Kruglyak, L. (2006). Genetics of global gene expression. *Nature Reviews. Genetics*, *7*(11), 862–872. doi:10.1038/nrg1964

Roth, R. B., Hevezi, P., Lee, J., Willhite, D., Lechner, S. M., Foster, A. C., & Zlotnik, A. (2006). Gene expression analyses reveal molecular relationships among 20 regions of the human CNS. *Neurogenetics*, *7*, 67–80. doi:10.1007/s10048-006-0032-6

Rzhetsky, A., & Gomez, S. M. (2001). Birth of scale-free molecular networks and the number of distinct DNA and protein domains per genome. *Bioinformatics (Oxford, England)*, *17*(10), 988–996. doi:10.1093/bioinformatics/17.10.988

Sasaki, M., Takeda, E., Takano, K., Yomogida, K., Katahira, J., & Yoneda, Y. (2005)... *Genomics*, *85*(5), 641–653. doi:10.1016/j.ygeno.2005.01.003

Scannell, D. R., & Wolfe, K. (2004). Rewiring the transcriptional regulatory circuits of cells. *Genome Biology*, *5*(2), 206. doi:10.1186/gb-2004-5-2-206

Schroeder, M. D., Pearce, M., Fak, J., Fan, H., Unnerstall, U., & Emberly, E. (2004). Transcriptional control in the segmentation gene network of *Drosophila*. *PLoS Biology*, *2*(9), E271. doi:10.1371/journal.pbio.0020271

Scott, M. P., & Carroll, S. B. (1987). The segmentation and homeotic gene network in early Drosophila development. *Cell*, *51*(5), 689–698. doi:10.1016/0092-8674(87)90092-4

Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U., & Gaul, U. (2008). Predicting expression patterns from regulatory sequence in Drosophila segmentation. *Nature*, *451*(7178), 535–540. doi:10.1038/nature06496

Sherlock, G., Hernandez-Boussard, T., Kasarskis, A., Binkley, G., Matese, J. C., & Dwight, S. S. (2001). The Stanford microarray database. *Nucleic Acids Research*, *29*(1), 152–155. doi:10.1093/nar/29.1.152

Stormo, G. D. (2000). DNA binding sites: Representation and discovery. *Bioinformatics (Oxford, England)*, *16*(1), 16–23. doi:10.1093/bioinformatics/16.1.16

Strogatz, S. H. (2001). Exploring complex networks. *Nature*, *410*, 268–276. doi:10.1038/35065725

Stuart, J. M., Segal, E., Koller, D., & Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic module. *Science*, *302*(5643), 249–255. doi:10.1126/science.1087447

Su, A. I., Wiltshire, T., Batalov, S., Lapp, H., Ching, K. A., & Block, D. (2004). A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, *10*(16), 6062–6067. doi:10.1073/pnas.0400782101

Teichmann, S. A., & Babu, M. M. (2004). Gene regulatory network growth by duplication. *Nature Genetics*, *36*(5), 492–496. doi:10.1038/ng1340

Tong, A. H. Y., Lesage, G., Bader, G. D., Ding, H., Xu, H., & Xin, X. (2004). Global mapping of the yeast genetic interaction network. *Science*, *303*(5659), 808–813. doi:10.1126/science.1091317

Tran, L. S., Nakashima, K., Shinozaki, K., & Yamaguchi-Shinozaki, K. (2007). Plant gene networks in osmotic stress response: From genes to regulatory networks. *Methods in Enzymology*, *428*, 109–128. doi:10.1016/S0076-6879(07)28006-1

Tsang, J., Zhu, J., & van Oudenaarden, A. (2007). MicroRNA-mediated feedback and feedforward loops are recurrent network motifs in mammals. *Molecular Cell*, *26*(5), 753–767. doi:10.1016/j.molcel.2007.05.018

Tsantoulis, P. K., & Gorgoulis, V. G. (2005). Involvement of E2F transcription factor family in cancer. *European Journal of Cancer*, *41*(16), 2403–2414. doi:10.1016/j.ejca.2005.08.005

Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(9), 5116–5121. doi:10.1073/pnas.091062498

van Dijk, A. D., ter Braak, C. J., Immink, R. G., Angenent, G. C., & van Ham, R. C. (2008). Predicting and understanding transcription factor interactions based on sequence level determinants of combinatorial control. *Bioinformatics (Oxford, England)*, *24*(1), 26–33. doi:10.1093/bioinformatics/btm539

van Driel, M. A., Bruggeman, J., Vriend, G., Brunner, H. G., & Leunissen, J. A. (2006). A text-mining analysis of the human phenome. *European Journal of Human Genetics*, *14*(5), 535–542. doi:10.1038/sj.ejhg.5201585

Velculescu, V. E., Zhang, L., Vogelstein, B., & Kinzler, W. (1995). Serial analysis of gene expression. *Science*, *270*, 484–487. doi:10.1126/science.270.5235.484

Vermeirssen, V., Barrasa, M. I., Hidalgo, C. A., Babon, J. A., Sequerra, R., & Doucette-Stamm, L. (2007). Transcription factor modularità in a gene-centered C. elegans core neuronal protein-DNA interaction network. *Genome Research*, *17*(7), 1061–1071. doi:10.1101/gr.6148107

Vidal, M., Braun, P., Chen, E., Boeke, J. D., & Harlow, E. (1996). Genetic characterization of a mammalian protein-protein interaction domain by using a yeast reverse two-hybrid system. *Proceedings of the National Academy of Sciences of the United States of America*, *93*(19), 10321–10326. doi:10.1073/pnas.93.19.10321

Wasserman, W. W., & Sandelin, A. (2004). Applied bioinformatics for the identification of regulatory elements. *Nature Reviews. Genetics*, *5*(4), 276–287. doi:10.1038/nrg1315

Watts, D. J., & Strogatz, S. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*, 440–442. doi:10.1038/30918

Wei, G. H., Liu, D. P., & Liang, C. C. (2004). Charting gene regulatory networks: Strategies, challenges, and perspectives. *The Biochemical Journal*, *381*(1), 1–12. doi:10.1042/BJ20040311

Wittkopp, P. J. (2005). Genomic sources of regulatory variation in cis and in trans. *Cellular and Molecular Life Sciences*, *62*(16), 1779–1783. doi:10.1007/s00018-005-5064-9

Wray, G. A. (2003). Transcriptional regulation and the evolution of development. *The International Journal of Developmental Biology*, *47*(7-8), 675–684.

Wu, W. S., Li, W. H., & Chen, B. S. (2006). Computational reconstruction of transcriptional regulatory modules of the yeast cell cycle. *BMC Bioinformatics*, *7*, 421. doi:10.1186/1471-2105-7-421

Yu, H., Kim, P. M., Sprecher, E., Trifonov, V., & Gerstein, M. (2007). The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. *PLoS Computational Biology*, *3*(4), e59. doi:10.1371/journal.pcbi.0030059

Zenke, M., & Hieronymus, T. (2006). Towards an understanding of the transcription factor network of dendritic cell development. *Trends in Immunology*, *27*(3), 140–145. doi:10.1016/j.it.2005.12.007

Zhang, S., Jin, G., Zhang, X. S., & Chen, L. (2007). Discovering functions and revealing mechanisms at molecular level from biological networks. *Proteomics*, *7*(16), 2856–2869. doi:10.1002/pmic.200700095

Zhao, Y., & Karypis, G. (2005). Data clustering in life sciences. *Molecular Biotechnology*, *31*(1), 55–80. doi:10.1385/MB:31:1:055

ADDITIONAL READINGS

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). *Molecular biology of the cell*. New York and London: Garland Science.

Babarasi, A. L. (2002). *Linked: The new science of networks*. MA: Perseus Publishing Cambridge.

Baldi, P., & Hatfield, G. W. (2002). *DNA microarrays and gene expression*. Cambridge, UK: Cambridge University Press.

Barabasi, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, *286*(5439), 509–512. doi:10.1126/science.286.5439.509

Barabasi, A. L., & Bonabeau, E. (2003). Scale-free networks. *Scientific American*, 50–59.

Bartel, D. P. (2004). MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell, 116*, 281–297. doi:10.1016/S0092-8674(04)00045-5

Bornholdt, S., & Schuster, H. G. *Handbook of graphs and networks: From the genome to the Internet.* Hoboken, NJ: Wiley/VCH.

Buck, M. J., & Lieb, J. D. (2004). ChIP-chip: Considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, *83*, 349–360. doi:10.1016/j.ygeno.2003.11.004

Dorogovtsev, S. N., & Mendes, J. F. F. (2003). *Evolution of networks: From biological networks to the Internet and WWW*. Oxford, UK: Oxford University Press.

Douglas, W. B. (2001). *Introduction to graph theory (2nd ed.)*. Upper Saddle River, NJ: Prentice Hall.

Gilbert, S. F. (2000). *Developmental biology*. Sunderland, MA: Sinauer Associates, Inc.

Han, J. D. (2008). Understanding biological functions through molecular networks. *Cell Research*, *18*(2), 224–237. doi:10.1038/cr.2008.16

Koonin, E. V., Wolf, Y. I., & Karev, G. P. (2006). *Power laws, scale-free networks, and genome biology*. New York: Springer U.S.

Lee, C. T., Risom, T., & Strauss, W. M. (2007). Evolutionary conservation of microRNA regulatory circuits: An examination of microRNA gene complexity and conserved microRNA-target interactions through metazoan phylogeny. *DNA and Cell Biology*, *26*(4), 209–218. doi:10.1089/dna.2006.0545

Li, H., Xuan, J., Wang, Y., & Zhan, M. (2008). Inferring regulatory networks. *Frontiers in Bioscience*, *13*, 263–275. doi:10.2741/2677

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., & Califano, A. (2006). ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics, 7*(Suppl 1), S7. doi:10.1186/1471-2105-7-S1-S7

Mason, O., & Verwoerd, M. (2007). Graph theory and networks in biology. *IET Systems Biology*, *1*(2), 89–119. doi:10.1049/iet-syb:20060038

Neal, S. J., & Westwood, J. T. (2006). Optimizing experiment and analysis parameters for spotted microarrays. *Methods in Enzymology*, *410*, 203–221. doi:10.1016/S0076-6879(06)10010-5

Newman, M., Barabási, A. L., & Watts, D. J. (2006). *The structure and dynamics of networks.* Princeton, NJ: Princeton University Press.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256. doi:10.1137/S003614450342480

Pastor-Satorras, R., & Vespignani, A. (2004). *Evolution and structure of the Internet: A statistical physics approach*. Cambridge, UK: Cambridge University Press.

Radicchi, F., Castellano, C., Lecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(9), 2658–2663. doi:10.1073/pnas.0400054101

Thomas, M. C., & Chiang, C. M. (2006). The general transcription machinery and general cofactors. *Critical Reviews in Biochemistry and Molecular Biology*, *41*(3), 105–178. doi:10.1080/10409230600648736

Vidal, M. (2001). A biological atlas of functional maps. *Cell*, *104*(3), 333–339. doi:10.1016/S0092-8674(01)00221-5

Watts, D. J. (2003). *Six degrees: The science of a connected age*. New York: W. W. Norton & Company.

Watts, D. J. (2003). *Small worlds: The dynamics of networks between order and randomness*. Princeton, NJ: Princeton University Press.

Wu, Z., Irizarry, R. A., Gentleman, R., Murillo, F. M., & Spencer, F. (2004). *A model based background adjustment for oligonucleotide expression arrays* (Tech. Rep.). John Hopkins University, Department of Biostatistics, 1001.

ENDNOTES

1 The Gene Ontology project (http://www.geneontology.org) provides a controlled vocabulary to describe genes and gene product attributes in any organism.

2 OMIM™ (http://www.ncbi.nlm.nih.gov/omim/) is a catalogue of human genes and genetic disorders.

Section 2

Network Inference

Chapter 3
Bayesian Networks for Modeling and Inferring Gene Regulatory Networks

Sebastian Bauer
Charité Universitätsmedizin Berlin, Germany

Peter Robinson
Charité Universitätsmedizin Berlin, Germany

ABSTRACT

Bayesian networks have become a commonly used tool for inferring structure of gene regulatory networks from gene expression data. In this framework, genes are mapped to nodes of a graph, and Bayesian techniques are used to determine a set of edges that best explain the data, that is, to infer the underlying structure of the network. This chapter begins with an explanation of the mathematical framework of Bayesian networks in the context of reverse engineering of genetic networks. The second part of this review discusses a number of variations upon the basic methodology, including analysis of discrete vs. continuous data or static vs. dynamic Bayesian networks, different methods of exploring the potentially huge search space of network structures, and the use of priors to improve the prediction performance. This review concludes with a discussion of methods for evaluating the performance of network structure inference algorithms.

INTRODUCTION

A multiplicity of mathematical tools has been developed to represent gene regulatory networks (GRNs) with different levels of detail. In the setting of network structure inference from microarray data, *Bayesian networks* (BNs) represent a commonly used tool to describe the network in a comparatively high level manner, in contrast, say, to ordinal differential equations. The purpose of this chapter is to provide necessary background knowledge of BNs.

The structure of this chapter is as follows: in the first section we provide a brief introduction into the biology of GRNs and the mathematical concepts on which the Bayesian networks are based. In

DOI: 10.4018/978-1-60566-685-3.ch003

the next section we present the theory of Bayesian networks and show how they can be adapted to *model* GRNs. We learn how we can use the model to *infer* or predict activity states of genes in terms of probability theory, which in general has been one of the classic uses of BNs. Yet, the probably most prominent application of Bayesian networks in computational biology has been for *reverse engineering* of gene regulatory networks, especially since the advent of high-throughput screening methods such as gene-expression microarrays. This is covered in the fourth section, in which we also discuss the issue of variable time lags in time-series data whereby the response time of one gene regulated by another varies greatly among the genes. We finish the chapter with conclusions, and provide directions which might be of interest for future research.

BACKGROUND

Biology

GRNs coordinate the changes in cellular behavior associated with development or response of the cell or organism to extracellular stimuli. Transcription factors are the molecules that activate or repress downstream genes by binding to promoter and other sequences (cis-regulatory modules) of genes, thereby modulating the rate of transcription of genes. Combinations of transcription factor binding events in any one promoter are one of the important factors determining the level of the corresponding mRNA in the cell. The regulatory state of the cell has been described as the total set of active transcription factors. However, a number of other molecules influence the activation state and concentration of transcription factors. For instance, signaling pathways consisting of ten or hundreds of proteins can transduce an extracellular event (such as the binding of a ligand to a receptor) into an intracellular biochemical signal by cascading protein modification events. For instance, a receptor-ligand binding event may induce phosphorylation (and activation) of an intracellular signaling molecule, which in turn phosphorylates other molecules, thereby propagating the signal through a cascade or network of proteins, some of which activate transcription factors and thereby influence the transcription of target genes. Other factors, such as non-coding RNAs, histone modifications, and CpG methylation, can also influence the level of mRNA of target genes. Therefore, measurement of mRNA levels can provide only a partial view of the regulatory state of a cell. At present, however, there remain major technical difficulties in obtaining large-scale measurements of protein levels or protein modifications, so that network structural inference has for the most part been attempted with mRNA data.

Graph Theory

Graphs are abstract entities of discrete mathematics which are used to encode relationships of interest between objects of the same domain. Formally, a *graph* is a pair $G=(V,E)$, in which V is finite set of *vertices*, representing the objects, and E a set of pairs of distinct elements of V, which is a binary relation over V. Elements of E are called *edges* (or arcs). The pairs may be ordered or not. An order implies a *direction*. If all edges of G are directed, the graph is *directed*. If at least one edge is directed we call the graph a *partially directed graph*. Otherwise the graph is an *undirected graph*.

A *path* with length n is a sequence of vertices $(v_1,...,v_n)$ which respects the edges, i.e., $(v_i, v_{i+1}) \in E$ for all i. A *cycle* is a special path whose start vertex v_1 equals to the end vertex v_n. A *directed path* is a

path, in which the edges between the vertices are all directed. A directed graph is *acyclic* if it contains no directed cycle; such graphs are referred to as *directed acyclic graphs* (DAGs). A *partially directed acyclic graph* (PDAG) is a graph which contains directed and undirected edges, but which doesn't contain any directed cycle.

Probability Theory

A *probability space* is a triplet (Ω,Σ,P), in which the *sample space* Ω defines all possible elementary events of an experiment. The set Σ contains events, based upon the sigma-algebra of subsets of Ω. The *probability measure P* maps any event $E \in \Sigma$ to a real value between 0 and 1 such that $0 \leq P(E) \leq 1$. In addition, a probability measure must satisfy $P(\Omega)=1$, and for any number of sequences of *n* disjoint events $P\left(E_1 \cup E_2 \cup \ldots \cup E_n\right) = \sum_{i=1}^{n} P\left(E_1\right)$.

A *random variable* is a function that maps elements from the *sample space* Ω to a measurable space, the *state space* (often real-valued). A *probability distribution* is a probability measure over the state space.

If the sample space of a random variable is finite or countable then the random variable is said to be *discrete*. The probability measure is then described by a *probability mass function* (pmf). As an example consider throwing a coin. The sample space is countable, it can be *Heads* or *Tails*, therefore Ω={*Heads*,*Tails*}. The state space could be {0,1}, mapping the outcomes to measurable entities, i.e., entities that we can calculate with. The pmf of such variables is a *Bernoulli distribution*, which, in this particular case, would assign to both elementary events 0.5 if the coin is fair.

Now consider a random experiment, in which every trial results in one of *k* possible outcomes, where the probability of observing an outcome *i* is given by p_i. When repeating this random experiment *m* times, let X_i count the number of times outcome *i* is observed. The pmf is then described by a *multinomial distribution* which is given by

$$P(X_1 = x_1,\ldots,X_k = x_k) = f(x_1,\ldots,x_k;p_1,\ldots,p_k) = \frac{\left(x_1 + \ldots + x_k\right)!}{x_1!\ldots x_k!} p_1^{x_1} \ldots p_k^{x_k}, \quad (1)$$

where $\sum_{i=1}^{k} x_i = m$. Note that for coin example we would have $k=2$, and $p_1=p_2=0.5$.

The concept of random variables can be extended to uncountable sets as well. A random variable *X* is said to be *continuous* if its probability distribution is continuous, i.e., it is a *probability density function* *f*(*x*), which is $f(x) \geq 0$ for all $x \in R$ and

$$\int_{-\infty}^{\infty} f(x)dx = 1.$$

The probability of $a \leq X \leq b$ denoted as $P(a \leq X \leq b)$ can be calculated by integrating the density function from *a* to *b*. Note that this implies that for continuous random variables $P(X=a)=0$, for all $a \in R$.

As it is in the discrete case, there are several common classes of continuous probability distributions. A very popular distribution for continuous variables is the *normal distribution*, also referred to as the *Gaussian distribution*. The density function of the Gaussian is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean and σ^2 the variance. The density function is often abbreviated as $N(\mu,\sigma^2)$. The *multivariate normal distribution* is a generalization of the normal distribution to more than one variable.

Another continuous distribution is the Dirichlet distribution. It is a multivariate distribution, whose density of order κ with parameter $\alpha_i > 1$ for $1 \le i \le \kappa$ is given by

$$f(x_1,\ldots,x_\kappa;\alpha_1,\ldots,\alpha_\kappa) = \frac{\Gamma\left(\alpha_1 + \ldots + \alpha_\kappa\right)}{\Gamma\left(\alpha_1\right)\ldots\Gamma\left(\alpha_\kappa\right)} x_1^{\alpha_1 - 1} \ldots x_\kappa^{\alpha_\kappa - 1} \tag{2}$$

assuming that $0 \le x_i \le 1$ and $\sum_{i=1}^{\kappa} x_i = 1$. The *gamma function* $\Gamma(\cdot)$ is a generalization of the factorial for real numbers $x \in R$, that is, $\Gamma(x+1) = x\Gamma(x)$.

For any probability space, two events, say *A* and *B*, are said to be *independent* if and only if $P\left(A \cap B\right) = P(A)P(B)$. The conditional probability of event *A* given *B* denoted by *P*(*A*|*B*) is defined as $\frac{P\left(A \cap B\right)}{P\left(B\right)}$. It represents the probability of *A* if it is known that *B* has occurred. If *A* and *B* are independent, it follows that *P*(*A*) = P(*A*|*B*). We say that *A* and B are *conditionally independent* given a third event *C*, if $P\left(A \cap B \mid C\right) = P(A \mid C)P(B \mid C)$. Two random variables *X* and *Y* are said to be *independent* if and only if any outcome of *X* is independent given any outcome of *Y*, denoted by *I*(*X*;*Y*). That is *X* and *Y* are independent in their probability distribution. *X* and *Y* are conditionally independent given another random variable *Z*, if they are independent given any outcome of *Z*. We denote this by *I*(*X*;*Y*|*Z*).

A *joint probability distribution* (jpd) is a probability distribution of two or more random variables together. The joint probability distribution of two variables *X* and *Y* is denoted by *P*(*X*,*Y*). The *marginal probability distribution* (mpd) of *X* is the probability distribution of *X* ignoring *Y* altogether. Depending whether *Y* is discrete or continuous, it can be determined by summarizing or integrating according to the probability distribution over *Y*'s state space. If the jpd consists of more than one other variable in addition to *X* then we summarize over all combinations of the states of the other variables, denoted by $\sum_{\neg\{X\}}$.

Bayes' Theorem

Bayes' theorem follows from the definition of the conditional probability and relates the conditional probability *P*(*A*|*B*) to *P*(*B*|*A*) for two events *A* and *B* such that

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P\left(B\right)}.$$

In this context, *P*(*B*|*A*) is referred to as the *likelihood*, as it is a probability of parameter *B*, in contrast to *P*(*A*|*B*) which called the *posterior*; it is derived from the knowledge of *B*. *P*(*A*) is referred to as the *prior*, as it represents the knowledge of *A* prior to the knowledge of *B*. *P*(*B*) is the *normalization constant*.

If the posterior $P(A|B)$ has the same algebraic form as the prior $P(A)$ then the prior is said to be the *conjugate prior* to the likelihood. For instance, if the likelihood is a multinomial distribution (Equation 1) and the prior is a Dirichlet distribution (Equation 2) then the posterior will also have a Dirichlet distribution, albeit with updated hyperparameters α_i. Therefore the Dirichlet distribution is a conjugate prior to the multinomial distribution.

BAYESIAN NETWORKS

Bayesian Networks can be seen as a mixture of graph theory and probability theory. A BN is pair $B = (G,\Theta)$ consisting of a directed acyclic graph $G=(V,E)$ and a set Θ of *local probability distributions* (LPDs).

The vertices (nodes) of the graph $V=\{1,...,n\}$ bidirectionally map to variables $X=\{X_1,...,X_n\}$. The directed edges in E stand for direct dependency relations of one variable to another. We say that X_i is a *parent* of X_j, if there is an edge from node i to node j. The set of indices of all parents of X_i is denoted by $pa(i)$. A family is defined as the set of a variable and all of its parents. For a concrete realization (assignment) of a set of variables we use the term *configuration*.

The DAG encodes independence relations following the *Markov condition*, which states that a variable given the parents doesn't depend on any other *non-descendants*, i.e., those variables to which no directed path exists.

In addition to the structural properties, for every $X_i \in X$ there is a local probability distribution (LPD) defined which depends only on the configuration of the parents denoted as $p(X_i|X_{pa(i)})$. As a variable given the configuration of the parents is independent to all other variables, the multidimensional joint probability of all variables can be calculated as:

$$p\left(X_1,\ldots,X_n \mid G\right) = \prod_{i=1}^{n} p\left(X_i \mid X_{pa(i)}\right). \tag{3}$$

DAGs encoding a certain conditional independence are not necessarily unique in the space of all DAGs. For example consider the following conditional independence relation: $I(Y;Z|X)$. All three conceivable Bayesian network structures for which this relation is true are shown in Figure 1.

In contrast, the first structure depicted in Figure 2 encodes quite a different independence relation: $I(Y;Z)$. Such structures, that is, subgraphs consisting of three nodes in which the edges of two nodes converges into the other one are referred to as *v-structures*.

In general we say that two DAGs are *equivalent* if they encode the same set of conditional independences. As proven in Pearl and Verma (1990) this is the case only for such graphs that have the same *skeleton*, which is constructed from a DAG by omitting the direction of the edges, and the same *v-structures*. The equivalence relation naturally imposes a set of equivalent classes onto the space of all DAGs. The equivalent classes can be represented uniquely by PDAGs. The second part of Figure 2 displays the PDAG for the example (which is a simple undirected graph here).

Figure 1. Three different DAGs that all encode I(Y;Z|X) and therefore belong to the same equivalence class

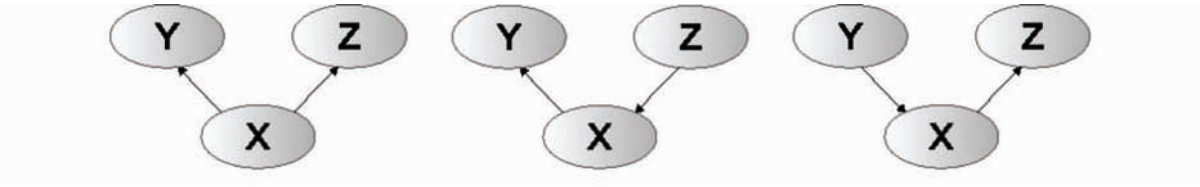
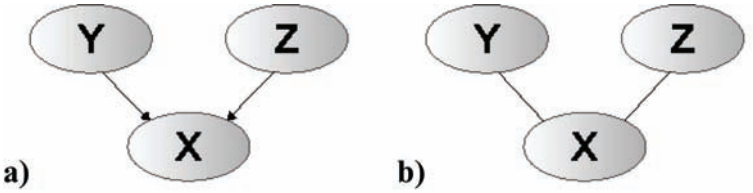


Figure 2. The first structure a) represents the I(Y;Z) but not I(Y;Z|X). The second b) represents the equivalence class of the three DAGs depicted in Figure 1.



TWO COMMON LPDS

Although any LPD can be used for BN analysis, two are extensively used in practice: the multinomial distribution (MD) for discrete variables and the normal (Gaussian) distribution (GD) for continuous variables.

The MD for a variable X_i with *m* discrete states is a function of all members of the variable's family which maps all possible configurations to a probability value between 0 and 1 such that for every parent configuration π_i

$$\sum_{j=1}^{m} p\left(X_i = j \mid X_{pa(i)} = \pi_i\right) = 1.$$

Usually, the MD is given as a conditional probability table (CPT).

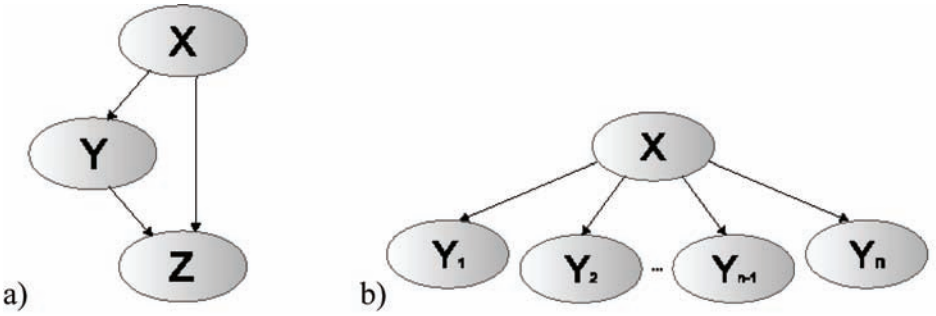
For the GD, the distribution for each variable follows a normal distribution whose mean depends linearly on the configuration of the parents:

$$p(X_i \mid X_{pa(i)}) = N(X_i, \mu_i + \sum_{j \in pa(i)} b_{ij}(X_j - \mu_j), \sigma_i^2),$$

where b_{ij} defines the strength of the influence of variable X_j on X_i. Note that $b_{ij} \neq 0$, otherwise one would not include X_j in the parent set of X_i.

Note that while non-linear relationships can be modeled using the MD, the fact that the mean of the GD is a linear function of the states of the parents means that non-linear relationships cannot be modeled with the GD. Also note that a BN is not required to have either discrete or continuous nodes. Instead one can mix nodes by defining different types of LPDs for the nodes.

Figure 3. Two common gene regulatory network motifs; a) feed-forward loop b) single-input module consisting of the master transcription factor X and n regulated genes



MODELING GENE REGULATORY NETWORKS

In order to model gene regulatory networks using the BN framework, the genes are mapped to corresponding random variables. The transcriptional regulations, i.e., activation or deactivation, are modeled intuitively by the edges in the graph such that there is an edge from every *regulator gene* to its *target gene*. A *transcriptional family* consists of a single target gene and all its regulator genes. The precise transcriptional influence within a transcriptional family is given using the LPD for the target gene.

A variety of network motifs have been described for gene regulatory networks. The *feed-forward loop* (FFL) is typical for transcriptional networks, and indeed is one of the most frequent motifs in sensory transcriptional networks consisting of three genes (Alon 2007). Let us name the genes X, Y, and Z. The characteristic of the motif is that gene X regulates gene Y and that gene Z is regulated by both X and Z. The FFL motif as depicted in Figure 3a can be easily mapped to a Bayesian network. The concrete influence, that is, whether an edge activates or deactivates the target gene can be modeled via the GD or MD. A synergistic effect, however, can be modeled with the MD only.

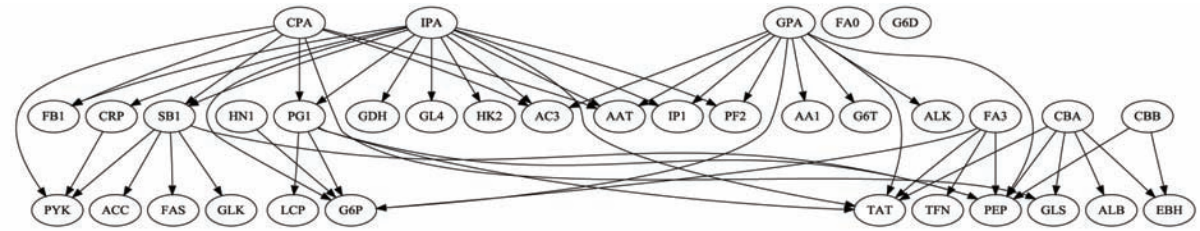
Another important motif is the *single-input module* (SIM) as depicted in Figure 3b. This motif is as common for both sensory and developmental transcription networks as the FFL is. The main feature is that there is a so-called master transcription factor which exclusively regulates a set of target genes in the same regulatory fashion (i.e., all of them are either activated or deactivated). As the activation thresholds of the target genes' transcription varies, often the SIM occurs when there is a need for kind of assembly line, in which the temporal order of the expression is important. The system which is responsible for the construction of the flagella in *E.coli* (Kalir et al. 2001) is a prominent example that employs such a motif. Although the principal relationships can be mapped easily to a BN, the characteristic time dependent properties (i.e. the order of the gene transcription) can hardly be modeled by this class of Bayesian networks, also referred to as *static Bayesian networks*.

For instance, Le et. al. (2004) constructed a network for the *hepatic glucose homeostasis*. The network contains 35 genes, some of which genes map to the insulin, glucagon, and glucocorticoid signaling pathways. Every gene is modeled as a discrete variable with two states representing low and high activity. For the construction of the relationships they used domain knowledge gained from intensive literature research resulting in 52 regulatory interactions. The graphical representation of their network is depicted in Figure 4. The CPT for gene EBH is given as an example in Table 1.

Table 1. Depicted is the CPT for gene EBH which is regulated by CBA (activating) and CBB (dominantly repressing). If both transcription factors CBA and CBB are low then the activity is modeled as 30%.

CBB	CBA	Low	High
Low (0)	Low (0)	0.7	0.3
Low (0)	High (1)	0.1	0.9
High (1)	Low (0)	0.9	0.1
High (1)	High (1)	0.9	0.1

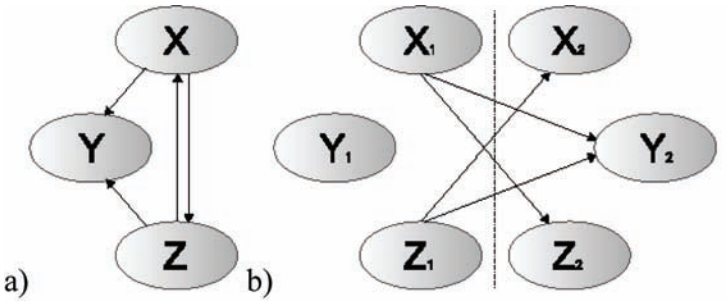
Figure 4. The structure of the Bayesian network of hepatic glucose homeostasis process as constructed by Le et al (2004)



A drawback of the Bayesian network approach is that it is not possible to model motifs which consist of a loop. In addition to the motifs described above, another common motif of gene regulatory networks is the *feed-back loop* (FBL), which often appears in developmental transcription networks. Here gene X and Z both regulate gene Y, but gene X and Z also regulate each other. The graphical representation is depicted in Figure 5a. However, as this graph is not acyclic, it is not a valid structure for a Bayesian network which requires the structural graph to be acyclic.

In real biological networks genes and therefore transcription factors are transcribed in a different amount at different rates depending on the process they are involved in. Furthermore, the threshold for an activity of a transcription factor depends not solely on its amount but also on the specific properties of the protein (e.g., its affinity to the DNA) and varies greatly. Thus, the regulation of one gene by another doesn't result in instantaneous changes of the expression level of the regulated gene. In fact, the time

Figure 5. The structure of the feed-back loop network motif. The network shown in panel a) cannot be represented as a BN because there is cycle. Unrolling the temporal relationships as shown panel b) leads to a valid dynamic BN.



required to transcribe a gene ranges from about 1 minute in bacteria and yeast to 30 minutes (including mRNA processing) in mammals (Alon 2007).

The common way to incorporate such time delays is to duplicate the set of variables as many times as discrete time steps need to be taken into account. The first set of variables is used to represent genes at time point *i*. The other sets are assigned to subsequent time steps, i.e., *i* + 1, *i* + 2, and so forth. Edges existing between these sets can be seen as a directed time-delayed regulation. These kinds of Bayesian networks are usually referred to as *dynamic Bayesian networks* (DBN). By unrolling a graph containing directed loops such as the mentioned FBL one can derive a dynamic Bayesian network as depicted in Figure 5b.

INFERENCE

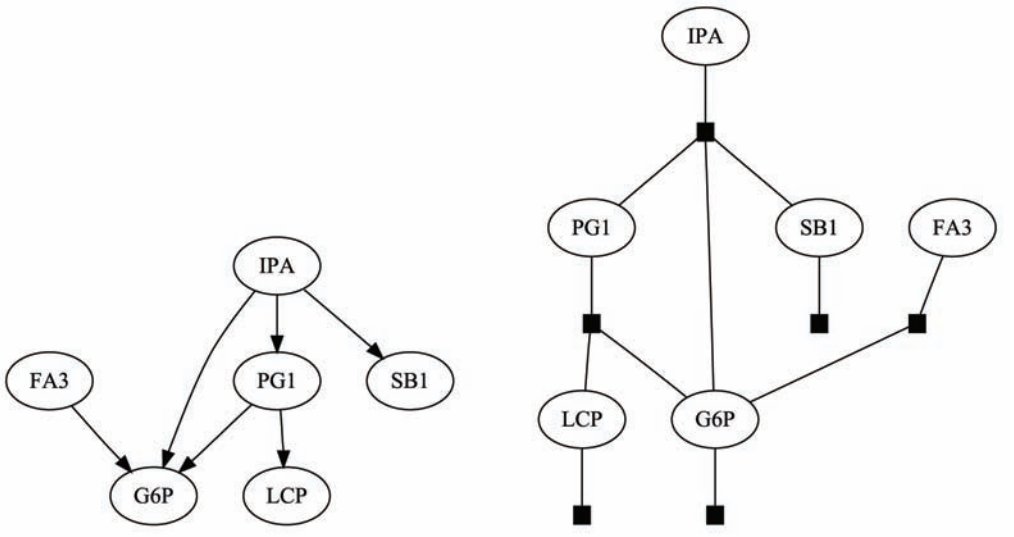
Using a fully specified BN instance, one can make predictions about an outcome given the states of a set of variables (the *evidence*). This is one of the main applications of the Bayesian networks. For example, consider that we know that the gene IPA is in the state *high* and that the gene G6P is in state *low*, which state is the most likely one for the gene LCP?

Recall that a BN is just a way to express a joint probability distribution, $P(\mathbf{X})$ with $\mathbf{X}$ being the vector of all random variables of the BN. Therefore we can answer such questions by calculating the marginal probability distribution of the respective objective variable, whereby all the observations (evidence) are incorporated during the calculations. In particular, for the question given above, we calculate $P(\mathrm{LCP}) = \sum_{\neg\{\mathrm{LCP}\}} P(\mathbf{X} \mid \mathrm{IPA} = high, G6P = low)$. At the first sight, this summary operation seems to be carried over a huge space because the network consists of a total of 35 genes. However, as the Bayesian network factorizes the joint probability distribution as a product of local distributions due to Equation 1 the problem becomes more easily solvable, after some rearrangements such as the exploitation of the distribution law. The procedure is commonly termed *belief propagation*.

A general algorithm for the so-called marginalize product-of-functions problem (MPF) is the *sum-product algorithm* which is applied on a *factor graph*. (Frey et al. 1998, Kschischang et al. 2001) A factor graph is an undirected graph which expresses how a global function is factorized using local functions, also referred to as the *factors*. It is a bipartite graph, in which the first set of vertices represents the variables, and the second set the factors. There is an edge between a variable vertex and a factor vertex, if and only if the factor depends on the variable. The derivation of a factor graph representing the same joint probability as a particular Bayesian network therefore can be easily achieved: Take twice as many nodes as there are variables in the BN; the first half represents the variables, the second represent the factors. Every factor corresponds to the LPD attributed to a variable. Then, for every random variable *X* draw an edge from its associated factor to those nodes that are located in the first set and represent the family members of *X*. Figure 6 illustrates the factor graph of a subnet of the *hepatic glucose homeostasis* network mentioned above.

The sum-product algorithm can be described in terms of a message passing algorithm acting on the factor graph. Here we consider an acyclic factor graph. The algorithm begins at the leaves of this factor graph, i.e. nodes that only have a single neighbor. Leaf variable nodes send a trivial identify function message to their neighbors, leaf factor nodes send a description of the function to their neighbors.

Figure 6. On the left side a subnet of the hepatic glucose homeostasis is depicted, while the corresponding factor graph is depicted on the right side. This graph encodes the structure of the joint probability density function with aid of two set of nodes: the variables (depicted as circles) and the factors representing the LPDs (depicted as squares). The edges between the nodes represent the variables' dependencies.



Each node *v* waits for a message from all but one of the adjacent nodes before it sends a message along the remaining edge to a node *w* based upon a computation according to its type: a variable node computes the product of the received factor messages, a factor node representing function *f* forms the product of *f* with the received variable messages and applies the summary operator $\sum_{\neg\{x\}}$ where *x* represents the variable of node *w*.

Node *v* then waits for a returning message from node *w.* Then it sends a new message back to all adjacent nodes except for *w*. The algorithm terminates as soon as for each edge one message for each direction has passed. Thus, by calculating the product of all incoming messages, all variable nodes x_i then have determined their corresponding marginal function.

BAYESIAN NETWORK LEARNING

In the previous section, we showed how Bayesian networks can be used to predict the outcome of an event, e.g., to find out the downstream result of a perturbation. Before one can apply any inference algorithm, of course, a meaningful model of the process in question is needed.

One possibility is to use expert knowledge to build a network for a gene regulatory process as it was done in the example of *hepatic glucose homeostasis*. This is practicable for smaller or less detailed networks, which are described in literature. Another possibility is to let the computer find statistical relations in observed data, as, for instance, obtained by microarray technology. These so-called learning procedures have an important role in research: they enable scientists to discover relations that have not been listed yet, which is the actual goal of molecular biology. One nice feature of the presented Bayesian network framework is that it unifies these two approaches, and as we will see, in a quite elegant fashion.

For Bayesian networks there are two kinds of properties which can be learned from observed data:

- the parameter of the local probability distribution, and
- the structure of the graph.

In the following we will concentrate on the latter method, as for a biologist this is the most interesting feature of GRNs, especially if only little about the studied GRN is known.

Learning the structure of a Bayesian network means that we like to find such a network that best explains the observed data. This can be done using a variety of approaches, e.g., by formulating the problem as an optimization problem.

The intuition behind optimization algorithms is that we use a scoring measure to evaluate the goodness of a single network with respect to the given data. By scoring every feasible network of the space of all networks we then take the one which scores best. However, this approach is bound to fail as the number of possible DAGs is super-exponential in the number of vertices as derived by Robinson (1973). Worse yet, Chickering (1996) showed that such a problem is NP hard with respect to the number of variables which essentially means that with current knowledge there is no algorithm known that is capable of finding an optimal solution in acceptable amount of time.

In such cases we can usually fall back to approximation algorithms, heuristics or stochastic procedures, which will be the topic of this section. This is also the part in which the gene-expression data comes into play. Due to the probabilistic nature of Bayesian networks such learning procedures have the advantage that they can deal with noisy data inherent to the microarray technology fairly well. Moreover, they also allow prior knowledge to be easily incorporated which can improve the ability to infer the correct network.

One important aspect of learning in general is the issue of *overfitting*. Intuitively, overfitting means that the learned model represents the training data too well. For instance, Bayesian networks whose structure is determined by a fully connected directed graph can surely explain more data than a less dense graph could do. What we ought to look for are structures that explain the data fairly well but avoid the model becoming too complex. This process, i.e., the process of balancing complex models against less complex models which may not entirely reflect all relationships, is termed *regularization*.

Let (G, Θ) be a Bayesian network as defined above. Furthermore, let D be the complete data consisting of c cases $D= (d_1,...,d_c)$ from which we want to learn the structure. In order to find a structure reflecting the observed data, one seeks a model, whose graph structure G maximizes $P(G \mid D)$. This reflects the application of a *maximum a posteriori* (MAP) approach, because according to the Bayes' theorem we have

$$P\left(G \mid D\right) = \frac{1}{Z} P\left(D \mid G\right) P\left(G\right), \tag{4}$$

where $P(D|G)$ is *marginalizedlikelihood*, and $P(G)$ in general the *prior*, in particular the *structural prior*. The divisor

$$Z = \sum_{G} P\left(D \mid G\right) P\left(G\right) \tag{5}$$

is referred to as the *normalization constant*. Because the model space is super-exponential in size calculating Z is not possible if the number of variables is large. In order to find a maximum it suffices to consider the product $P(D|G)P(G)$. If no prior knowledge is available then the uniform distribution is assigned to $P(G)$, i.e., every model is equal likely. In this special case it suffices to maximize $P(D|G)$. The marginalized likelihood $P(D|G)$ is the result of the integration of the likelihood with respect to the parameter prior over the whole parameter space Θ

$$P\left(D \mid G\right) = \int P\left(D \mid G, \theta\right) P\left(\theta \mid G\right) d\theta. \tag{6}$$

It can be shown that, if the data is complete, this integral becomes analytically solvable for certain probability distributions of the likelihood, especially for the multinomial distribution and the normal distribution, when a conjugate prior is used.

This averaging and weighting according to the parameter prior amounts to Occam's razor, because overly complex network models with many free parameters are penalized. This is intuitively explained in Riggelsen (2006): consider a dense graph and sparse graph and note that the models are different in the number of free parameters: for the dense graph, more parameters need to be determined than for the sparse graph. Therefore the distribution of the parameter prior $P(\Theta|G)$ of the dense graph has a flatter shape than the distribution $P(\Theta|G)$ of a sparse graph. As the density of every point is smaller, complex structures are penalized.

Let *score*(D,G) be a function which assigns the graph G with respect to data D a certain real number. An important property of the scores that we will consider in the following is the property of *decomposability*. That is, in order to calculate the score of a graph it suffices to calculate the score of every family. The score of the graph is then composed by determining the product of all these scores denoted as:

$$score\left(D,G\right) = \prod_{i=1}^{n} score\left(D_{i,pa(i)}, X_i, X_{pa(i)}\right)$$

Previously, we have already noticed that more than one DAG graph may capture the same conditional independence. While this doesn't impose a problem if an expert defines the network structure as he defines the causal relation, we cannot distinguish DAGs from other DAGs belonging to the same score equivalence class from data alone. Therefore algorithms which learn from *observational* data alone can merely produce PDAGs. In order to learn causal relations we somehow have to fix a variable and re-initiate the experiment which generated the data. In the setting of learning GRNs this means that we have to perturb a gene's expression, for instance, by doing a knockout or overexpression study, and apart from that repeat the experiment under the same conditions. In their study Werhli et al. (2006) showed that the ability to correctly detect edges increased significantly.

Note that the case is different if we consider learning from time course data as the casual relations are defined by the time. Yet, perturbations also help here to uncover the regulatory relationships as the purpose of perturbations is changing the dynamics.

Note that even though we can only distinguish score equivalent classes most algorithms operate on the space of DAGs although an operation on the smaller space of PDAGs seem to be more suitable. This however is mainly due the simplicity of the natural operations in the space of DAGs. In contrast local operations in the space of all PDAGs are more complicated.

Discrete Scoring Metrics

As already noted above, the conjugate prior for the multinomial is the Dirichlet distribution. For x_i let $n(x_i,x_{pa(i)})$ be the number of configurations with $X_i=x_i$ and $X_{pa(i)}=x_{pa(i)}$ within the data. Furthermore, let $n\left(x_{pa(i)}\right)=\sum_x n\left(x,x_{pa(i)}\right)$ be the number of configurations for which x is marginalized out. For the discrete case, the likelihood can be determined using

$$P\left(D \mid G\right)=\prod_{i=1}^{n}\prod_{x_{pa(i)}}\frac{\Gamma\left(a\left(x_{pa(i)}\right)\right)}{\Gamma\left(a\left(x_{pa(i)}\right)+n\left(x_{pa(i)}\right)\right)}\prod_{x_i}\frac{\Gamma\left(a\left(x_i,x_{pa(i)}\right)+n\left(x_i,x_{pa(i)}\right)\right)}{\Gamma\left(a\left(x_i,x_{pa(i)}\right)\right)}$$

where a is the prior belief of the certain configuration following from the $P(\theta \mid G)$ of Equation 6. It could be given by an expert for a particular G, but obviously this is impracticable. Rather than specifying the parameter for every model we can select a single probable model, say G', along with its parameters, say θ'. We then let

$$a\left(x_i,x_{pa(i)}\right)=ESS\cdot P\left(x_i,x_{pa(i)} \mid G',\theta'\right),$$

where *ESS* is the so-called *equivalent sample size*. Both parameters contribute to the regulation. *ESS* represents, as the name suggests, the magnitude of the belief in the prior, i.e., to how many samples have been already seen on which the prior is founded. While therefore the ESS can be attributed to the global regulation, the factor $P(x,x_{pa(i)}|G', \theta')$ amounts to the local regulation, that is, the regulation for every vertex. The score is then referred to as the *Bayesian Dirichlet equivalent* (BDe).

The specification of such a probable G' and θ' is often not possible. If we let G' be the empty graph and assign a uniform distribution to $P(x_i|G',\theta')$ we get what is termed *Bayesian Dirichlet equivalent uniform* in literature. This eventually leads to

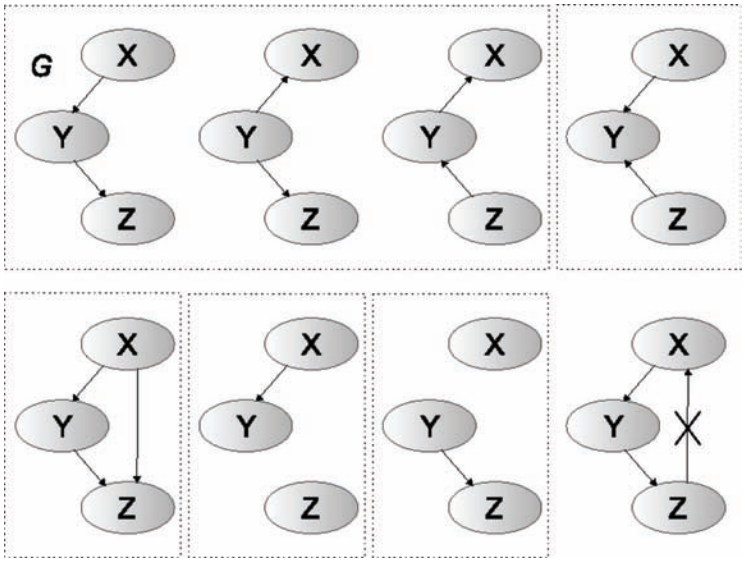
$$a(x_i,x_{pa(i)})=\frac{ESS}{\left|\Omega(x_i)\right|\cdot\prod_{j\in pa(i)}\Omega(x_j)}$$

where a $|\Omega(x_i)|$ gives the number of discrete states of variable x_i.

The K2 score simply assigns all $a(x_i,x_{pa(i)})$ a value of 1. Its main drawback is that it is not score equivalent in contrast to the previous scores. Therefore, it is not really suited when learning the model solely from data. However, the K2 score initially was derived as part of the K2 algorithm (Cooper et al., 1992) which assumes that the order of the nodes is known. In this setting the criterion of score equivalence is not relevant (Riggelsen 2006).

A disadvantage when applying a discrete score is that data has to be available in discrete form. This may require a discretization step beforehand, and thus an associated loss of information. But then it may also lead to the reduction of noise. A basic method to discretize the continuous microarray data was applied by Friedman and co-workers (2000). Based upon a control, they assigned three states to the values, depending on whether they are overexpressed, underexpressed, or equally expressed. Others take the mutual information of the genes' expression values into account with the objective to preserve it as much as possible as done by Hartemink and co-workers (2002).

Figure 7. All possible structures that can be derived by applying a rule to the graph G. Graphs which are placed in the same box represent the same equivalence class. The last graph of the bottom row is not in the neighborhood as the proposed change would make it cyclic. The neighborhood N(G) therefore contains six graphs that can be obtained from G by single operations.



Continuous Scoring Metrics

Analogous to the multinomial case a score for the case, in which the LPD of nodes follows a normal distribution, a continuous scoring metric was derived by Geiger (1994). This score is referred to *Bayesian Gaussian equivalent* (BGe) score. The advantage of using this score is that the data doesn't need to be discretized, bypassing a potential information loss. The disadvantage is again that non-linear dependencies cannot be detected.

Strategies for Finding a Good Model

As noted in the first part of the section, a brute force attempt to find the optimal model can only succeed for relatively small sized networks. So far, conceiving an efficient algorithm is also not possible unless P=NP. Several heuristics to find good fitting models have been proposed. In the following we will introduce the most widely used algorithms.

One of the general approaches to find a good solution of an optimization problem is the so-called greedy hill-climbing algorithm. Given an instance of a model we systematically perform small *local* changes to the model in order to find that modification that increases the objective score at most. Using this model we repeat the procedure until no other change can produce a model which fits the optimization criteria better. With this algorithm, only a local optimum can be found.

In the setting of Bayesian networks the rules for local changes that can be applied encompass adding an edge between two nodes, removing an edge, or switching an edge. Given a fixed structure G we call the set $N(G)$ the neighborhood of G, which encompasses all DAGs that can be derived from G by the application of a single operation, as depicted in Figure 7.

Technically, because of the scores' decomposability, it is enough to recalculate the scores of the affected families to weight the change. One further important result of studies of gene regulatory networks or biological networks in general is that the underlying graph is sparse. Applied to the Bayesian networks framework this means that the size of a transcriptional family is relatively small, i.e., the number of parents of a gene doesn't exceed a certain constant, say *k*. Such gene is said to have a maximum fan-in of *k*.

An early adopter of this observation is the so-called Sparse Candidate Algorithm (SCA) of Friedman et al. (1999). SCA is a variant of the greedy algorithm. Every iteration *j* can be divided into two phases:

1. In the *restriction phase*, we select for every variable X_i a candidate set C_i with $|C_i| \leq k$ which defines all potential parents of X_i. This induces another graph: $H^j = \left(\left\{X_1, \ldots, X_n\right\}, \left\{\left(X_i, X_j\right) | \forall i, j : X_j \in C_i\right\}\right)$.
2. In the *maximization phase* we enumerate all possible acyclic subgraphs of H^j in order to find the graph G^j, which maximizes the optimization score.

The algorithm stops, either if score(G^j)=score(G^{j-1}) and if $H^j = H^{j-1}$ or if score(G^j)=score(G^{j-1}) for a certain number of iterations. A crucial aspect of the procedure is the way the candidates are selected in the restrict phase. One requirement is that set of candidates of parents for a variable always includes the current parents. Therefore for each iteration we get at least a structure which is as good as the previous one, ensuring the termination of the algorithm. Furthermore, a new parent variable is chosen for each variable X_i based upon the current parent set of X_i, whereby the candidate parent is selected whose inclusion into the family leads to the highest improvement of the score. Previous candidates of X_i which are not present in the parent set are discarded.

From an algorithmic point of view we note that the detection of directed loops doesn't come for free. When using a static algorithm this can take up to $O(n + m)$ steps where *n* is the number of nodes and *m* the number of edges in the graph. The cycle detection test is necessary for every operation which possibly can construct a cycle, i.e., the adding or reversing of an edge. For the SCA this would be performed in the maximization phase. Thus, rather than applying a static algorithm for detecting cycles it makes sense to consider using a dynamic algorithm such as the one of Marchetti-Spaccamela et al.(1996) or of Katriel and Bodlaender (2006). The property of these algorithms is that they maintain the topological order of nodes while edges are inserted and removed. The problem of topological sorting is related to directed cycle detection.

When we want to infer the structure of a dynamic Bayesian network, however, the test for acyclicity can be omitted: edges may be only directed from an earlier time point to a later time point. The cardinality of the neighborhood is always the same.

MCMC in the Space of DAGs

When inferring the network structure from microarray data, the data usually is sparse, which means that the number of available samples is relatively small compared to the number of variables (genes). The probability distribution $P(G|D)$ is then expected to have a wide shape, so no single network has a clear maximum score. Rather than that there can be networks whose scores are close to each other.

This suggests the idea to sample networks according to the posterior $P(G|D)$, which essentially means that good models have higher chance to be sampled than models which explain the data poorly. From this set of sampled networks, interesting features can be extracted as formalized in Friedman et al. (1999). For instance, one could construct a weighted DAG in which the edges are weighted according to the

number of their occurrences in sampled DAGs. This way, a kind of ranking about the confidence of the edges can be obtained. Also more complicated features such as counting particular network motifs are conceivable. However, at first glance the sampling doesn't seem to be possible due to the normalization constant in (4).

A solution to this problem was brought up by Madigan and York (1995) who adapted the Metropolis-Hastings algorithm for process of learning Bayesian networks. As for the context of finding gene regulatory networks it has been for instance used by Husmeier (2003). First note that in general a time-homogeneous Markov chain is stochastic process over the space of models G_i which is defined as

$$P_{n+1}\left(G_i\right) = \sum_k T\left(G_i \mid G_k\right) P_n\left(G_k\right),$$

where $T(G_i \mid G_k)$ determines the probability of going into state G_i given that we are at state G_k. Under fairly mild conditions this converges to a stationary distribution P_∞ uniquely defined by T from which we can sample from just by running the chain.

For learning structural properties of Bayesian networks the model space is made up of all possible DAGs with a fixed number of vertices. As above a transition is a small local change that leads to a graph in the neighborhood $N(G_k)$. All we have to do is to define T such that the stationary distribution of the Markov chain P_∞ equals the posterior $P(G|D)$, that is:

$$P\left(G_i \mid D\right) = \sum_k T\left(G_i \mid G_k\right) P\left(G_k \mid D\right).$$

But this is the case if the so-called equation of *detailed balance* holds:

$$\frac{T\left(G_k \mid G_i\right)}{T\left(G_i \mid G_k\right)} = \frac{P\left(G_k \mid D\right)}{P\left(G_i \mid D\right)}.$$

Usually $T(G_k \mid G_i)$ is composed as a product of a proposal probability $Q(G_k \mid G_i) = |N(G_i)|^{-1}$and an acceptance probability $A(G_k \mid G_i)$. The intuition behind this is that first we randomly select a new structure following the proposal distribution and then accept it corresponding to the acceptance probability. The acceptance probability is determined as

$$A(G_k \mid G_i) = \min\left\{\frac{P(G_k \mid D)Q(G_i \mid G_k)}{P(G_i \mid D)Q(G_k \mid G_i)}, 1\right\},$$

which, after applying Bayes' theorem, becomes

$$A(G_k \mid G_i) = \min\left\{\frac{P(D \mid G_k)P(G_k)Q(G_i \mid G_k)}{P(D \mid G_i)P(G_i)Q(G_k \mid G_i)}, 1\right\}.$$

This means that one can also plug-in an arbitrary score and incorporate prior knowledge. Note that, if a uniform structural prior is assumed, then these priors can be cancelled out. Also note that, in case of a dynamic Bayesian network, the proposal distribution can also be cancelled out, as the neighborhoods are all equal in size.

Using these properties, it is now possible to formulate the algorithm to sample the networks. First, we start with an arbitrary initialized network (e.g., an empty graph), say G_0. Then, for i=1 to *N*, we

- randomly select a structure G_i^p from the proposal distribution $Q(G_i|G_{i-1})$
- accept the new model, i.e., $G_i = G_i^p$, with probability $A(M_i \mid M_{i-1})$.

Before one can consider the G_i as proper samples, i.e., before the Markov chain reaches its stationary distribution, the chain has to be run several steps, though. This phase is also referred to as the *burn-in* phase of the Markov chain. This value and therefore the parameter *N* can go into the thousands or ten of thousands, but this highly depends on the size of the network and on the data. Often it makes sense to monitor the acceptance probability: if it is subject to high fluctuations, it is very unlikely that the chain has reached the equilibrium. In order to determine the confidence, it is also helpful to repeat the run for several times using different initialization settings.

MCMC in the Space of the Orders

So far, when we scored networks we dealt with fully specified network structures based upon DAGs. Another approach is to forget the structure for a moment and concentrate in a first step on the topological order of the *n* nodes as suggested by Friedman and Koller (2000). The authors showed how one can compute the posterior of network structures using this order. Furthermore, it is possible to apply other algorithms that benefit from this information in the construction of the true structure, such as the already mentioned K2 algorithm. But it could be also interesting per se, as the order can indicate genes upstream the regulation process, for example, providing feasible candidates for perturbation experiments.

Denote by *O* the order of the nodes. Analogous to the previous section, we want to sample from the posterior $P(O|D)$. In order to do so, we construct a Markov chain which consists of all *n*! possible orders such that the Markov chain has the stationary distribution $P(O|D)$. Denote by $Q(O'|O)$ the probability of moving from *O* to *O'*. This could involve flipping the order of two randomly selected nodes, i.e., we change the order $(i_1,...,i_j,...i_k,...i_n)$ to $(i_1...i_k...i_j...i_n)$. We accept the proposal with probability of

$$A(O' \mid O) = min\left\{\frac{P\left(O' \mid D\right)Q\left(O \mid O'\right)}{P\left(O \mid D\right)Q\left(O' \mid O\right)}, 1\right\}.$$

Notice that according to Bayes' theorem for any *O* we have

$$P\left(O \mid D\right) = \frac{P\left(D \mid O\right)P\left(O\right)}{P\left(D\right)}, \text{ and thus } \frac{P\left(O' \mid D\right)}{P\left(O \mid D\right)} = \frac{P\left(D \mid O'\right)P(O')}{P\left(D \mid O\right)P(O)}.$$

The relation of P(O) against P(O') can be again neglected, if no prior information is available, i.e., all orders are equally possible. The likelihood $P(D|O)$ can be indeed calculated in a closed form as given by Friedman and Koller (2000).

$$P\left(D \mid O\right) = \prod_i \sum_{j \in U_{i,O}} score\left(D_{i,j}, X_i, X_j\right).$$

where $U_{i,O}$ contains the sets of all parents, which can precede X_i in the given Order O. The cardinality of the elements of $U_{i,O}$ can be restricted to not exceed a certain size of k, meaning that the maximum fan-in of gene represented by X_i is k.

Structural Priors

The term $P(G)$ allows the user to favor some models over others to the extent that graphs with certain edge configurations, i.e., that lack of or feature a particular edge, are assigned a higher probability. This way, information can be integrated which is not derivable by microarray data alone.

Sequence based properties of the involved genes can be seen as one source of a prior. That is, if a promoter of a gene X is predicted to have binding sites of a product of another involved gene Y it can be assumed that the final network consist an directed edge from Y to X. The existence of such predicted relationships is used to build a prior graph. A simple but rigid incorporation of such links would be to attribute a probability of zero to those structures which lack the links which are in the prior graph as for instance done in Hartemink et al. (2002). A procedure which adheres to the noisy nature of the predicted binding sites was used in Husmeier (2003): basically for every agreement between the prior graph and G on an edge the prior of G is weighted by a value of $\varphi>1$. A more involved method is due to Segal et al. (2002).

Other priors involve limitations on the maximum fan-in of genes or favor net conformations such that genes encoding interacting proteins are more likely to be regulated by a common transcription factor. The Bayesian framework is quite flexible in this respect, and it is possible to incorporate almost any kind of biological information into the prior.

Time Lags

In the setting of inferring the network structure from a time series, for which one can use the DBNs, the incorporation of knowledge about time lags can improve the quality of the network considerably. Hence an important issue of any inferring algorithm applied on time series is the capability to detect the time a transcription factor needs to influence the transcription of its target gene in order to take the full advantage of the data.

In the following we briefly describe a method which can be used to determine the time lags. The method is due to Zou and Conzen (2005). They define that a gene j may be regulated by another gene i if its expression values changes after the value of gene i changed. Gene i is then called a *potential regulator* of j.

The first part of the procedure involves finding all potential regulators of any gene whereby a gene's expression is considered as changed when it reaches a certain threshold. The biological relevant transcription time lag between a regulator i and its target gene j is defined as the difference between the time points of initial expression change of i and j. The second part of the algorithm determines for every potential regulator-target pair the time lag according to this definition. This information is then used to set up the dynamic Bayesian network consisting of multiple time points.

Assessing Performance

Of course, running any algorithm on any data will always produce some models. But how much reality is really reflected in such model?

A major difficulty in assessing the performance of algorithms for reverse engineering GRNs is the fact that our knowledge about such networks is far from complete. Therefore, in most cases no gold standard is available against which the results of network structure inference could be compared. However, it is possible to use literature-derived regulatory interactions for comparison of results (e.g., Zou and Conzen, 2005). For example, typically one asks how many known interactions are identified as edges in the inferred network structure. Caution is needed, though, simply due to the fact that if an edge between two genes is predicted by the reverse engineering algorithm but was not previously known in the scientific literature, it may not be possible to distinguish between a lead towards a new discovery (the actual goal of the analysis!) and a false-positive prediction.

Another approach was performed by Friedman and coworkers (2000), taking advantage of the bootstrap method to generate multiple "perturbed" versions of the original dataset which still are reasonable models of the data, performing network inference, and determining the proportion of experiments in which a feature such as an edge between two genes is identified. In one experiment, the authors showed that analysis using a multinomial model on randomized data did not identify any feature in over 80% of the bootstrapped trials. They concluded that features identified in a greater proportion of trials using the original data were unlikely to represent mere artifacts.

A number of groups simulated GRNs to generate data for Bayesian network inference. In this case, since the structure of the "true" network is a given, it is possible to calculate the specificity and sensitivity of structural inference methods. Multiple approaches for simulating data have been proposed. One simple method is to construct a complete Bayesian network to reflect either known or synthetic networks (Husmeier 2003; Le et. al 2004; Geier et al. 2007). Then, as the structure as well as the parameters is known, we sample data according to this model. We then apply the inferring algorithm to this data. By determining the agreement of the resulting model to the original one we can assess the quality of the algorithm in certain respects, for instance ability to find the correct network subject to the number of data samples or to the amount of prior information.

Several groups used more realistic modeling procedures to generate synthetic network data. One approach is due to Zak and co-workers (2001). A small sized network described by chemical reactions consisting of a certain amount of genes, and includes transcription factor binding, transcription, translation, as well as protein-protein interaction events. Following the reaction-rate approach, these reaction equations can be shaped into a set of ordinary differential equations (ODEs). These can be integrated using an arbitrary initial value to obtain a function of the concentration against time for every involved species.

One then can imitate a typical microarray experiment, in which merely the abundance of the mRNA species can be measured, by only considering the mRNA profiles. One selects certain time points as one would in the real experiment. One can then feed the data obtained from the ODE model to the inference algorithm. As above we compare the resulting model with the original model in aspects of our choice.

In one elegant experiment, Husmeier (2003) adapted the network constructed by Zak et al. (2001) to include an additional 41 unconnected genes which were up- and down-regulated at random, and attempted to infer the original network structure from the simulated data using a DBN approach. Although

it was not possible to recover all true edges without false positive edges, the results did suggest that Bayesian network analysis could be used to make searching for novel genetic interactions significantly more effective than a search from *tabula rasa* (Husmeier 2003).

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Bayesian networks provide an intuitive way to describe relationships in the settings of gene regulatory networks and have become a popular tool for attempts at reverse engineering GRNs. For instance, Zhu and co-workers (2006) predicted a link between glucose repression and the YHB1 gene, which they verified by subsequent experiments.

An important open question for Bayesian network analysis and also other procedures for network structure inference is how to be measure performance of the algorithms and thus determine which algorithms are superior for application in biology. Current biological knowledge is far from complete; For instance, it has been estimated that only about 10% of all human protein-protein interactions are known (Hart et al., 2006). Therefore we think it is very important to develop modeling techniques using techniques such as ODEs, the stochastic Master equation, or hybrids, to develop systematic and realistic benchmarks. Other areas of research likely to be fruitful include the development of methods to take the different time courses of different biochemical reactions into account, i.e., better ways of capturing dependencies over multiple time points of a series of experiments. As new forms of high-throughput data become available (for instance, genome-wide binding data resulting from ChIP-Chip experiments), it will be important to incorporate this knowledge into appropriate priors.

REFERENCES

Alon, U. (2007). *An introduction to systems biology. Design principles of biological circuits*. London: Chapman & Hall.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning, 9*, 309–347.

Friedman, N., Goldszmidt, M., & Wyner, A. (1999). *Data analysis with Bayesian networks: A bootstrap approach.*

Friedman, N., & Koller, D. (2003). Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning, 50*, 95–126. doi:10.1023/A:1020249912095

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology, 7*, 601–620. doi:10.1089/106652700750050961

Hart, G. T., Ramani, A. K., & Marcotte, E. M. (2006). How complete are current yeast and human protein-interaction networks? *Genome Biology, 7*, 120. doi:10.1186/gb-2006-7-11-120

Hartemink, A. J., Gilford, D. K., Jaakkola, T. S., & Young, R. A. (2002). Combining location and expression data for principled discovery of genetic regulatory network models. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 437–449.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics (Oxford, England), 19*, 2271–2282. doi:10.1093/bioinformatics/btg313

Kalir, S. (2001). Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria. *Science, 292*, 2080–2083. doi:10.1126/science.1058758

Katriel, I., & Bodlaender, H. L. (2006). *Online topological ordering*.

Pearl, J., & Verma, T. S. (1990). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence* (pp. 220-227).

Riggelsen, C. (2006). *Approximation methods for efficient learning of Bayesian networks*. Unpublished doctoral dissertation, Utrecht University.

Robinson, R. W. (1976). Counting unlabeled acyclic digraphs. In *Proc. Fifth Australian Conf. Combinatorial Math*. (pp. 28-43).

Segal, E., Barash, Y., Simon, I., Friedman, N., & Koller, D. (2002). From promoter sequence to expression: A probabilistic framework. In *Proc. Sixth Annual Inter. Conf. on Computational Molecular Biology (RECOMB)*.

Werhli, A. V., Grzegorczyk, M., & Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models, and Bayesian networks. *Bioinformatics (Oxford, England), 22*, 2523–2531. doi:10.1093/bioinformatics/btl391

Zak, D. E., Doyle, F. J., Gonye, G. E., & Schwaber, J. S. (2001) Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data. In *Proc. 2nd Intl. Conf. Systems Biology*, 231-238.

Zak, D. E., Gonye, G. E., Schwaber, J. S., & Doyle, F. J. III. (2003). Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: Insights from an identifiability analysis of an in silico network. *Genome Research, 13*, 2396–2405. doi:10.1101/gr.1198103

Zhu, J., Jambhekar, A., Sarver, A., & DeRisi, J. (2006). A Bayesian network driven approach to model the transcriptional respone to nitric oxide in Saccharomyces cerevisiae. *PLoS ONE, 1*(1), e94. doi:10.1371/journal.pone.0000094

Zou, M., & Conzen, S. D. (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics (Oxford, England), 21*(1), 71–79. doi:10.1093/bioinformatics/bth463

ADDITIONAL READING AND RESOURCES

A comprehensive list of software for Bayesian networks can be found at http://www.csse.monash.edu.au/bai/book/appendix_b.pdf.

A fairly complete introduction to the theory of Bayesian networks is provided in the book "Learning Bayesian Networks" by Richard E. Neapolitan published by Prentice Hall in April 2003.

A nice free software package for both Bayesian network construction and inference is *GeNIe*, developed by the Decision Systems Laboratory group at the University of Pittsburgh. It is available from http://genie.sis.pitt.edu/ and requires a Windows operating system.

The homepage of our group dealing with gene regulatory networks is located at http://compbio.charite.de/genereg/. For instance, we provide the model of *hepatic glucose homeostasis* process ready to be fed into Genie.

With *WinMine*, which is a collection of tools rather than a single program, you can create models of Bayesian networks from discrete data. It is available from http://research.microsoft.com/~dmax/winmine/tooldoc.htm and requires a Windows operating system.

KEY TERMS AND DEFINITIONS

Bayesian Networks: A Bayesian network is a probabilistic graphical model. It contains of a graph whose vertices represent variables, for instance random variables. The directed edges of the graph encode direct dependency relation of one variable to another. Bayesian networks can be used to predict the state of variables, when other variables are fixed. In addition, Bayesian networks can be learned from sampled data.

Bayesian Scoring Metrics: A Bayesian Scoring Metric is a function that scores how well a given graph explains given data.

MCMC: The MCMC (Markov chain Monte Carlo) is a procedure which allows sampling instances from complex probability distribution. With respect to GRNs MCMC is used to sample from the space of all DAGs whereby the sampling scheme follows a distribution that is based on a Bayesian scoring metrics. Thus more probable DAGs, that is, DAGs that may better explain the data, are sampled more often and therefore one can construct a likely network structure.

Priors: A prior can be specified during a learning procedure that takes advantage of Bayes' theorem and may represent properties that are already known and therefore don't need to be rediscovered. It is especially useful when data is sparse, which is the case in micro array analysis, as it can significantly reduce the space of all DAGs that is used during the search.

Sparse Candidate Algorithm: The SCA is an approximation algorithm for the problem of finding a structure of a Bayesian network that maximizes a given Bayesian scoring metrics. It employs the feature that biological networks are usually sparse and consists of two phases, the restriction and the maximization phase.

Chapter 4
Inferring Gene Regulatory Networks from Genetical Genomics Data

Bing Liu
Monsanto Co., USA

Ina Hoeschele
Virginia Polytechnic Institute and State University, USA

Alberto de la Fuente
CRS4 Bioinformatica, Italy

ABSTRACT

In this chapter, we review the current state of Gene Regulatory Network inference based on 'Genetical Genomics' experiments (Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002; Jansen, 2003; Jansen & Nap, 2001; Schadt et al., 2003) as a special case of causal network inference in 'Systems Genetics' (Threadgill, 2006). In a Genetical Genomics experiment, a segregating or genetically randomized population is DNA marker genotyped and gene-expression profiled on a genomewide scale. The genotypes are regarded as natural, multifactorial perturbations resulting in different gene-expression 'phenotypes', and causal relationships can therefore be established between the measured genotypes and the gene-expression phenotypes. In this chapter, we review different computational approaches to Gene Regulatory Network inference based on the joint analysis of DNA marker and expression data and additionally of DNA sequence information if available. This includes different methods for expression QTL mapping, selection of regulator-target pairs, construction of an encompassing network, which strongly constrains the network search space, and pairwise and multivariate methods for Gene Regulatory Network inference, such as Bayesian Networks and Structural Equation Modeling.

DOI: 10.4018/978-1-60566-685-3.ch004

INTRODUCTION

A fruitful abstraction of biochemical systems is that of 'networks' (Barabasi & Oltvai, 2004; Dorogovtsev & Mendes, 2003; Newman, 2003; Pieroni et al., 2008; Watts & Strogatz, 1998). Such networks include Transcription Regulatory Networks (TRNs) (Lee et al., 2002; Luscombe et al., 2004; Shen-Orr, Milo, Mangan & Alon, 2002), Protein Interaction Networks (Pieroni et al., 2008; Schwikowski, Uetz & Fields, 2000), Metabolic Networks (Jeong, Tombor, Albert, Oltvai & Barabasi, 2000; Wagner & Fell, 2001), Gene Regulatory Networks (GRNs) (Brazhnik, de la Fuente & Mendes, 2002; D'Haeseleer, Liang & Somogyi, 2000) (see also A. de la Fuente – this book}, and Phenotype Networks (Nadeau et al., 2003). Inferring, or 'reverse engineering', such biological networks is therefore currently an area of research receiving a lot of interest and attention. It advances our knowledge about the integrated biochemical machinery of living cells (systems biology) and our understanding of general features of complex traits (complex trait biology). Constructing phenotype networks provides information about the functionality of complex systems (such as cardiovascular function) at the organismal level, and constructing GRNs furthers our understanding of the molecular basis of complex traits and diseases (Chen et al., 2008; Lum et al., 2006; Schadt et al., 2005). GRNs have other applications (Brazhnik, de la Fuente & Mendes, 2002), including the discovery of direct drug targets (di Bernardo et al., 2005; Gardner, di Bernardo, Lorenz & Collins, 2003). It has been shown that classical concepts from genetics, such as dominance and epistasis, can be readily understood in terms of networks and their properties (Kacser & Burns, 1981; Omholt, Plahte, Oyehaug & Xiang, 2000).

Many different experimental and computational approaches to GRN inference have been proposed. Data from experiments without targeted perturbations, or data from observational studies, only allow for inference of undirected Co-Expression Networks that are based on a measure of association between the expression profiles of pairs of genes (e.g.de la Fuente, Bing, Hoeschele & Mendes, 2004; Ghazalpour et al., 2006; Schäfer & Strimmer, 2005a,, 2005b; Wille & Buhlmann, 2006; Wille et al., 2004; Zhang & Horvath, 2005). In particular, one can construct an Undirected Dependency Graph (UDG), which contains edges only between those genes that interact directly, and which can be estimated based on partial correlations (de la Fuente, Bing, Hoeschele & Mendes, 2004; Shipley, 2002). The construction of a UDG can be a first step in a regulatory network analysis of a Genetical Genomics or Systems Genetics experiment.

A strategy of targeted perturbation is required to enable causal inference needed for the identification of the directed structure of GRNs. In such a strategy, targeted perturbations are created and responses of the gene-expression levels to the perturbations are measured. It has been shown that this approach can provide a reliable identification of GRNs (Brazhnik, de la Fuente & Mendes, 2002; de la Fuente, Brazhnik & Mendes, 2002; Gardner, di Bernardo, Lorenz & Collins, 2003; Wagner, 2001). There are two major types of targeted perturbation experiments. One approach uses one-at-a-time, specific perturbations in the expression of individual genes (e.g.Hughes et al., 2000; Mnaimneh et al., 2004). These experimental perturbations are relatively expensive and difficult to perform, especially in quantities required for comprehensive GRNs identification. Such perturbations (knock-outs, over-expressions) also tend to have strong biological effects, making it potentially difficult to distinguish between 'normal' functional relationships and relationships that emerge when the 'normal' functionality of a system is compromised.

The second type of targeted perturbation experiments, Genetical Genomics and Systems Genetics, uses naturally occurring, multi-factorial perturbations in segregating or genetically randomized populations

(Jansen, 2003; Jansen & Nap, 2001). Genetical Genomics is also referred to as 'the genetics of gene-expression' (Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002; Emilsson et al., 2008; Schadt et al., 2003), while Systems Genetics is defined more generally as the integration and anchoring of multi-dimensional data-types to underlying genetic variation (Threadgill, 2006). Genetical Genomics approaches integratively analyze gene-expression data and genotype data (measurable DNA sequence polymorphisms) and make use of DNA sequence information when available. DNA sequence polymorphisms that are identical to or genetically closely linked with some of the measured polymorphisms have been found to be an important source of gene-expression variation (e.g.Brem & Kruglyak, 2005), and hence they are the main reason why we can establish cause-effect relations. Multi-factorial perturbations offer an important advantage: "Any conclusion … has a wider inductive basis when inferred from an experiment in which the quantities of other ingredients have been varied …" (Fisher, 1954).

In this chapter, we review the current literature on GRN inference based on Genetical Genomics experiments and we indicate directions for further research.

BACKGROUND

In a Genetical Genomics experiment, a population for genetic mapping (a 'mapping population') consisting of hundreds of individuals is expression profiled for (ten) thousands of genes and genotyped for hundreds to thousands of genetic markers (measurable DNA polymorphisms). In yeast (Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002), plants (Keurentjes et al., 2007; West et al., 2007) and animal model systems (*e.g.* mouse) (Bystrykh et al., 2005; Schadt et al., 2003), such populations can be created by crossing two or more inbred strains and include backcrosses, Recombinant Inbred Lines (RILs), intercrosses, double haploids, etc.. For humans (Goring et al., 2007) and some farm animals, creating such crosses is not feasible, but existing segregating populations can be used, including large pedigrees and collections of 'unrelated' individuals. The variation in the expression levels of genes in a segregating population is influenced by the variation (genotypes) in many DNA polymorphisms across the genome (*e.g.* microsatellites, single nucleotide polymorphisms (SNPs) or Single Feature Polymorphisms (SFPs) to be discussed later). Establishing causal links between the genotype at each marker and one or more phenotypes of interest is known in genetics as Quantitative Trait Locus (QTL) mapping (Darvasi, 1998; Lander & Schork, 1994). QTL mapping identifies chromosomal regions (QTLs) that causally affect a phenotypic trait under consideration. Statistically, a QTL is a confidence interval for the genomic location of a DNA polymorphism that is causal for the phenotype of interest. This confidence interval is typically 1 to 20 centi Morgans (the unit of genetic distance whose relationship to physical distance varies between organisms) in length and hence can contain tens to hundreds of candidate causal polymorphisms. Because in Genetical Genomics the gene-expression levels are considered as phenotypic traits, the identified QTLs are referred to as 'expression-QTLs' or 'eQTLs'. Similarly, in the remainder of this chapter we will refer to gene-expression levels as 'expression traits' or 'etraits'.

Currently, mainly two Genetical Genomics datasets have been analyzed in the literature, yeast (Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002) and mouse (Schadt et al., 2005), but additional Genetical Genomics datasets have already been created for *C. elegans* (Li et al., 2006b), *A. thaliana* (DeCook, Lall, Nettleton & Howell, 2006; Keurentjes et al., 2007; Vuylsteke, van Eeuwijk, Van Hummelen, Kuiper & Zabeau, 2005; West et al., 2007), fruit fly (Anholt et al., 2003), human (Cheung et al., 2003; Cheung et al., 2005; Goring et al., 2007), and soybean (Zhou et al., 2008).

The widely used yeast data were created by crossing two genetically diverse yeast strains (Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002; Yvert et al., 2003). For a population of 112 haploid offspring, the gene-expression levels of 5736 genes and the genotypes of 2956 genetic markers were measured (Brem & Kruglyak, 2005). The yeast marker map in this study is very dense; for 90% of adjacent markers fewer than 10 recombinations occurred (Storey, Akey & Kruglyak, 2005). Brem et al. (Brem & Kruglyak, 2005) found that 3,546 gene-expression levels have a heritability of higher than 69%, meaning that 69% or more of their variance can be explained by genotypic variation. The median contribution to heritable expression variability by of a single identified QTL was 27%, and only 23% of all etraits were affected by a single QTL that explained more than half of the genetic variance, indicating that most expression traits are under the control of multiple polymorphisms (Brem & Kruglyak, 2005). The marker genotypes thus can be seen as naturally occurring 'genetic perturbations' responsible for (at least a large part of) the variation in gene-expression levels. In Figure 1 we present a nice illustration of the procedure for creating and using Genetical Genomics data, which we borrowed from the review on eQTL mapping by (Rockman & Kruglyak, 2006).

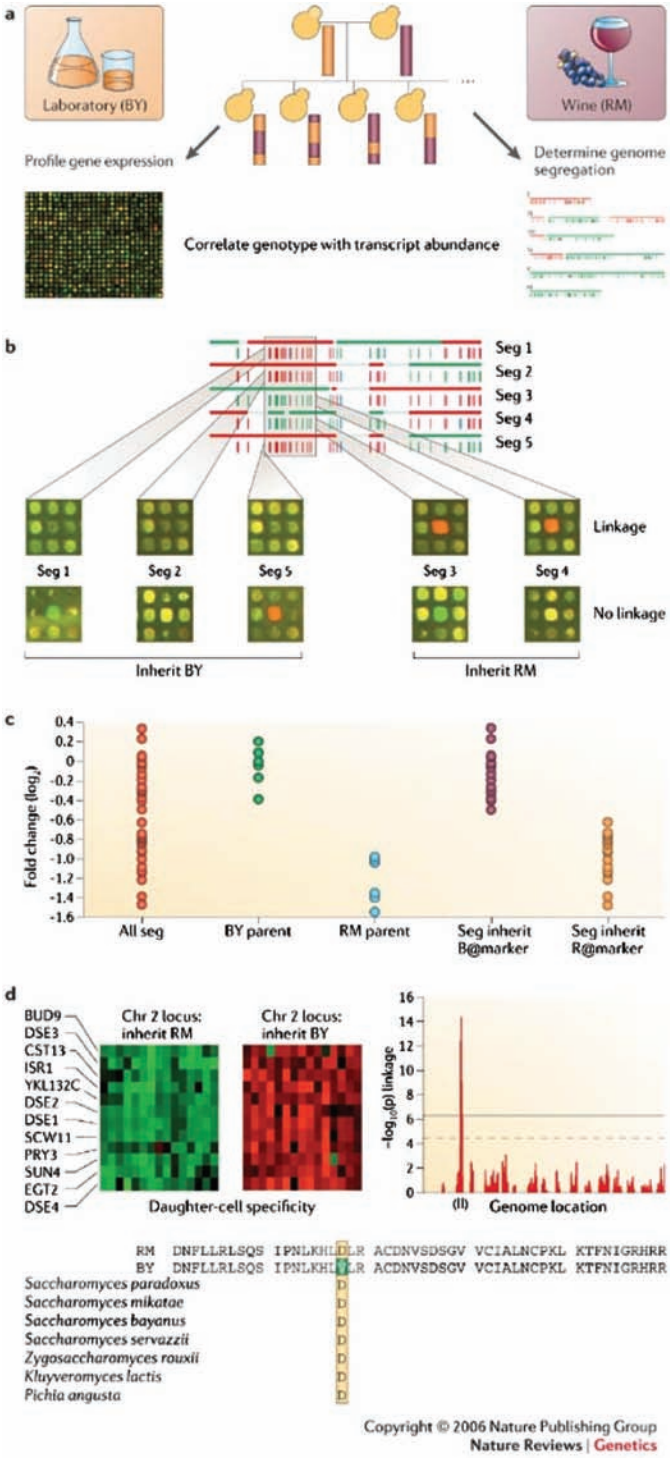
The result of eQTL mapping (see below for different approaches) is the knowledge that certain genomic regions likely have causal effects on the expression levels of particular genes. Then, genes located in an eQTL region can be identified as candidate regulators that are potentially responsible for the observed causal effects of the eQTL on the affected etraits. Since eQTLs can have wide confidence intervals, there may be many candidate regulators in a single eQTL. For the purpose of candidate regulator selection, several approaches have been proposed, including partial correlations (Bing & Hoeschele, 2005), between-strain SNPs followed by selection using Bayesian Networks (Li et al., 2005) and multiple-regression tests (Liu, de la Fuente & Hoeschele, 2008). The eQTL analysis and the candidate regulator identification provide strong constraints on the space of all possible GRNs underlying the data. The final task in inferring GRNs from Genetical Genomics data is to search for one or more optimal GRN structures within the constrained search space. Bayesian Networks have been used for this purpose (Lum et al., 2006; Zhu et al., 2004; Zhu et al., 2007). Bayesian networks use partially directed graphical models to represent conditional independence relationships among variables of interest and are suitable for learning from noisy data (*e.g.* microarray data) (Pearl, 2000; Spirtes, Glymour & Scheines, 1993). Unfortunately, Bayesian Networks are acyclic by definition and can thus not discover important feedback processes occurring in GRNs. Recent papers point to the need for methods that can infer cyclic networks, note the limitation of the Bayesian network approach (Lum et al., 2006) (see also de la Fuente – this book), and show better performance of a linear regression method over a Bayesian network algorithm most likely due to the presence of cycles (Faith et al., 2007). Therefore, Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) use a network model selection approach based on Structural Equation Modeling (SEM), which is related to Bayesian Network analysis, but it can model cyclic networks.

The approach summarized above for GRN inference consists of three steps: 1) eQTL mapping, 2) candidate regulator selection, and 3) refinement of the network structure. Below we will discuss each of the steps in detail.

EXPRESSION-QTL (EQTL) MAPPING

eQTL mapping is a major component of GRN inference in Genetical Genomics experiments. The quality of the network inference (measured for example as the false positive and false negative rates for the edges in the network) thus depends critically on the eQTL mapping accuracy.

Figure 1. The experimental design for a cross between two yeast strains; B: At a given genomic location, the samples are separated according to the inherited marker alleles, and linkage is declared if the groups differ significantly in expression; C: an actual linkage from a yeast cross; D: The eQTLs can be detected using molecular genetics tools. (From Rockman and Kruglyak, 2006. Reprinted with permission from the Publisher.)



Single-etrait-Single-eQTL Approach

The most straightforward approach to eQTL mapping is to use existing QTL mapping methods that have been devised for the analysis of a single or of a small number (say 2 to 20) of correlated phenotypes. The simplest method is to test the effect of each genetic marker in marker analysis (or of each candidate QTL position in interval mapping; see (Doerge, 2002; Doerge, Zeng & Weir, 1997) for reviews of these mapping methods) on each etrait individually. This method has been applied by several authors (e.g.Bing & Hoeschele, 2005; Brem & Kruglyak, 2005; Li et al., 2005; Liu, de la Fuente & Hoeschele, 2008; Yvert et al., 2003; Zhu et al., 2004) usually in combination with a significance threshold obtained by making adjustments for multiple testing based on the false discovery rate (FDR) control (e.g.Benjamini & Hochberg, 1995; Storey & Tibshirani, 2003).

This approach can produce large confidence intervals (in particular due to the presence of multiple linked QTLs), which can be at least partially remedied by using sliding three-marker regression (Thaller & Hoeschele, 2000) or composite interval mapping (CIM) (Jansen, 1993; Zeng, 1993; Zeng, 1994). Achieving the smallest possible confidence intervals is important to minimize the number of candidate gene regulators in each QTL region. Moreover, this approach has limited power, due to the large number of tests involved and the fact that pairwise relationships may not be strong enough and higher order relationships may need to be evaluated (although at the present time the evidence for the presence of interactions (epistasis) among eQTL is not strong). The simplest QTL (or here eQTL) mapping method, single marker regression, is based on the model

$$y_{in} = b_{i0} + b_{ij}x_{jn} + \varepsilon_{in} \tag{1}$$

where y_{in} is the etrait value for gene i and individual n of the segregating population, x_{jn} the genotype code for marker j and individual n, b_{i0} represents a mean expression value of gene i in the segregating population, b_{ij} is the (additive) effect of marker j on etrait i, and ε_{in} is a residual etrait value not explained by the effect of the marker. Based on this model a statistical test is performed to determine whether the marker effect b_{ij} is nonzero. This analysis is repeated for every marker j in a set of markers covering the genome. With this analysis, when there is an eQTL located on chromosome c, then the effect of every marker j located on chromosome c may be found to be nonzero. When using sliding three-marker regression or composite interval mapping, model (Eq. 1) is expanded to also include the effects of two markers whose genome positions flank the position of marker j, while only the effect of marker j is tested. Then, the effect of marker j is expected to be nonzero only if an eQTL is located between the two flanking markers, allowing for a more precise determination of the position of eQTLs in particular when there are multiple eQTLs on the same chromosome (when there are multiple eQTL on the same chromosome and only one marker is fitted in the model, then it is well-known that estimates of the eQTL position, i.e. determination of the marker nearest to an eQTL, can be biased). The choice of the flanking markers for each marker j is not a trivial task and requieres a compromise between maintaining sufficient power to detect a true effect of marker j and sufficient proximity of the flanking markers to marker j to minimize bias in the estimated eQTL position.

To determine which markers have nonzero effect b_{ij} on any etrait, one must choose a significance threshold by accounting for multiple testing across genes (etraits) and markers (eQTL positions). The False Discovery Rate (FDR) (Benjamini & Hochberg, 1995) has been a popular criterion for multiple testing control in standard QTL analyses and in eQTL analyses. However, the use of this criterion in

eQTL analysis is problematic, as described by Chen and Storey (Chen & Storey, 2006), essentially due to the strong correlation in signal among all marker tests on a chromosome containing at least one eQTL. Keurentjes *et al*. (Keurentjes et al., 2007), analyzing a Genetical Genomics dataset from an *Arabidopsis* RIL population, determined by simulation that achieving FDR control near the 0.05 level actually required using a more stringent (0.01) threshold. Alternatively, researchers have chosen to control the Family-Wise-Error Rate (FWER) (often referred to as the genome-wise error rate in the context of QTL mapping) separately for each etrait (by estimating adjusted, genome-wise p-values using data permutation (Churchill & Doerge, 1994)), and then to apply FDR control across the etraits (e.g.Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002; Chesler et al., 2005; Hubner et al., 2005). We applied this more conservative approach recently to a Genetical Genomics experiment with 300 RILs, 28395 etraits and 941 markers. We retained either the single top marker, or the top two, top three or top five markers for each etrait, and we applied FDR control to the resulting list of 1×28935, 2×28935, 3×28935 and 5×28935, respectively, genome-wise p-values. For an FDR level of 0.05 (0.01), we identified 21361 (21361), 31719 (30026), 35328 (23313), and 23453 (16970) eQTLs. These results show that including more than 2 or 3 candidate eQTL per etrait in the FDR control step actually led to a reduction in the total number of eQTL identified, indicating the limited power of this approach.

Multiple-etrait-Single-eQTL Approach

In standard multiple trait QTL mapping, the effect of each marker or QTL position on a set of correlated traits is evaluated. Multi-trait mapping can be more powerful than single trait mapping for detecting pleiotropic QTLs (e.g.Jiang & Zeng, 1995). However, this approach is computationally more demanding even for small numbers of traits and it is infeasible for (ten) thousands of etraits. It has been shown that using a small number of 'PC traits' (obtained by Principal Component Analysis of the original traits) is very effective for QTL mapping, when the original traits are (highly) correlated (in groups of traits). The PC traits are uncorrelated and can therefore be analyzed individually, and essentially the same QTL are identified by single trait analyses of few PC traits and by multi-trait analysis of the original traits (Jiang & Zeng, 1995; Mahler et al., 2002; Mangin, Thoquet & Grimsley, 1998). Therefore, the correlated nature of the large number of etraits can be utilized by deriving a much smaller number of (approximately) uncorrelated composite etraits. Several groups have used Principal Component Analysis, Hierarchical Cluster Analysis and K-means clustering individually or in combination to define composite etraits used to identify eQTLs with pleiotropic effects (Boomsma, 1996; Comuzzie, Mahaney, Almasy, Dyer & Blangero, 1997; Lan et al., 2003; Liu, de la Fuente & Hoeschele, 2008; Zeng et al., 2000).

Another approach to utilize correlated etraits when performing eQTL mapping for a particular etrait was suggested by (Pérez-Enciso, Quevedo & Bahamonde, 2007), proposing to include other etraits as covariates in the model. They showed that this approach can increase the power of eQTL identification and that the presence of other etraits in the model can strongly affect the results - some eQTL positions may be shifted, some eQTL may disappear, and some new eQTL may appear. These authors use information criteria for variable selection but other approaches could be used as well.

Multiple-eQTL Approaches Using Information Across Etraits

The methods described above are essentially applications of standard QTL mapping methods to eQTL mapping. The method of Kendziorski *et al*. (Kendziorski, Chen, Yuan, Lan & Attie, 2006) is probably

the first method specifically designed for eQTL mapping, but it has the disadvantage of mapping at most one eQTL per etrait. The power of eQTL mapping can be increased by simultaneous mapping of two or more eQTL, due to a reduction in the residual variance of the multiple QTL model, and because the multiple QTL model can incorporate the effects of interactions among eQTL that may sometimes be more important than their main effects. Multiple eQTL analysis can be performed by a multi-dimensional search, by simultaneously fitting the effects of all markers (or eQTL positions), or by a conditional or sequential approach, where a single QTL is identified first for a given trait, followed by a search for a second QTL based on a model including the already identified QTL and any second candidate QTL. An extension of these methods to the large number of etraits is not trivial and must utilize information across all etraits to be as powerful as possible. Storey *et al*. (Storey, Akey & Kruglyak, 2005) proposed a sequential method for identifying up to two eQTL per etrait and compared it with a complete two-dimensional search. With this method, some eQTL were identified that were not found with the single-etrait-single-marker approach (Brem, Storey, Whittle & Kruglyak, 2005; Storey, Akey & Kruglyak, 2005).

Jia and Xu (Jia & Xu, 2007) proposed a Bayesian model that allows for multiple eQTLs and utilizes information across transcripts. Their Bayesian method uses a well-known mixture prior distribution that explicitly models the null (zero effect) and alternative (non-zero effect) hypotheses for the effect of each marker on each etrait. It is essentially an extension of a well-known Bayesian variable selection method called Stochastic Search Variable Selection (McCulloch, 1996), which has previously been applied to QTL mapping (e.g.Yi, George & Allison, 2003, 2004), to eQTL mapping with a large number of etraits. However, Lucas *et al*. (Lucas, Carvalho, Wang, Bild & West, 2006) propose a modified mixture prior that may better account for sparsity in the analysis of a very large number of etraits. There are also non-Bayesian shrinkage methods for variable selection, including the Lasso (Tibshirani, 1996) and the Elastic Net (Zou & Hastie, 2005).

Multiple-etrait-Multiple-eQTL Approaches Based on Dimension Reduction

Methods that model individual etrait – eQTL associations are expected and have been found to have relatively low power, as stated earlier, and can be improved by utilizing information across etraits and fitting multiple markers or eQTLs simultaneously. However, such methods are computationally demanding and might still miss markers or eQTLs having fairly weak but consistent effects on a group of etraits that are also jointly affected by several other markers. Canonical Correlation Analysis (CCA) is a well-known multivariate statistical method that assumes two sets of normally distributed variables and finds a linear combination of the original variables in the first set and another linear combination in the other set that have the maximum correlation among all linear combinations. This pair of linear combinations is the first pair of canonical variates. Additional pairs of canonical variates that are maximally correlated after the previously identified pairs are also determined such that canonical variates from different pairs are uncorrelated.

Application of this classical CCA to the two sets of variables representing the etraits and the markers (or eQTL candidate positions) is not straightforward for several reasons. First, calculation of the canonical variates requires the estimation of the covariance matrices within and between sets, but the standard estimator of these covariance matrices fails because sample size is usually much smaller than the number of variables in each set, requiring some type of regularization. Secondly, the marker variables are discrete rather than normally distributed. Third, classical CCA is well-known to overfit small datasets, and hence a good tool for selecting the number of canonical variate pairs and for avoiding spurious correlations is

needed. Fourth, the results of classical CCA would be difficult to interpret, as all variables contribute to the linear combinations (as for PC analysis applied to the entire set of etraits (Liu, de la Fuente & Hoeschele, 2008)). Recently, several modifications of the classical CCA to identify associations among a set of etraits and a set of markers have been proposed (Beyene et al., 2007; Parkhomenko, Tritchler & Beyene, 2007; Waaijenborg, Verselewel de Witt Hamer & Zwinderman, 2008) that overcome some of the stated problems by using penalized versions of CCA. Further adaptations of CCA may be obtained based on a probabilistic interpretation of CCA (e.g.Wang, 2007). Other dimension-reduction methods for two sets of variables exist, but they have not yet been applied to Genetical Genomics, to our knowledge.

eQTL Mapping Using Sequence Information

The availability of sequence information implies that we know the physical location of the markers and the expression profiled genes. This knowledge allows us to perform eQTL mapping much more effectively by taking into account two distinct types of genetic regulation: cis- and trans-regulation. In the case of cis-regulation, a cis-eQTL affects a particular etrait X and is located at the physical location of gene X (the gene coding for etrait X) on a chromosome. The polymorphism of this cis-eQTL likely corresponds to a promoter region polymorphism in gene X (Doss, Schadt, Drake & Lusis, 2005; Jansen, 2003; Jansen & Nap, 2001; Liu, de la Fuente & Hoeschele, 2008; Rockman & Kruglyak, 2006; Ronald, Brem, Whittle & Kruglyak, 2005). The eQTL that cis-affects etrait X will have an indirect effect on the expression of those genes that are regulated by gene X (Doss, Schadt, Drake & Lusis, 2005). Such indirect effects have been referred to as cistrans effects (Kulp & Jagalur, 2006; Liu, de la Fuente & Hoeschele, 2008). Trans-eQTLs influence the expression levels of genes, but do not need to be co-located with any of these genes. A trans-eQTL likely is a coding region polymorphism in a regulator gene (Jansen & Nap, 2001; Liu, de la Fuente & Hoeschele, 2008; Rockman & Kruglyak, 2006; Yvert et al., 2003). While a trans-eQTL does not affect the expression level of the regulator gene, the coding region polymorphism affects the kinetic properties of the protein encoded by the regulator gene, which in turn affects the expression levels of the target genes. Since by definition the location of a cis-eQTL must physically coincide with the location of the gene whose etrait is affected, only the marker(s) closest to the location of an etrait's gene are tested to detect cis-eQTLs (Carlborg et al., 2005; Doss, Schadt, Drake & Lusis, 2005; Ronald, Brem, Whittle & Kruglyak, 2005). For network inference, finding cis-linked etraits by itself is not very informative. However, as shown on mouse data (Doss, Schadt, Drake & Lusis, 2005), the secondary targets of the cis-eQTLs, or the 'cistrans' regulated etraits, can be obtained by testing the effects of the identified cis-eQTL regions on all other etraits.

Trans-regulated target etraits are affected by both the eQTL genotype and the etrait of the regulator gene simultaneously. Therefore, it was proposed (Kulp & Jagalur, 2006; Liu, de la Fuente & Hoeschele, 2008) that, in order to specifically detect trans-eQTLs, in addition to the eQTL effect, the etrait of an associated candidate regulatory gene should be included as a covariate in the mapping model. In this way, eQTL mapping and regulator-target pair identification are incorporated in one step. Kulp and Jagalur performed interval mapping for any etrait *i* with a model including the effects of another etrait *j*, the effect of an eQTL at the physical location of gene *j*, and the etrait-by-eQTL interaction (Kulp & Jagalur, 2006). Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) performed trans-eQTL mapping by also including the etrait covariate of a candidate regulator gene associated with a candidate trans-eQTL, but they used marker regression and performed an Intersection-Union-Test (IUT) (Casella & Berger, 1990; Roy, 1957) to determine whether the eQTL genotype and the etrait of the candidate regulator gene both

significantly affect the target etrait, in whch case a trans-regulation was declared. Analyzing the yeast dataset of Brem and Kruglyak (Brem & Kruglyak, 2005), it was found that the etrait-by-eQTL interaction was rarely significant and essentially ignorable. While Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) found that this form of trans-eQTL mapping considerably increased the power of eQTL mapping, (Mancosu et al., 2008) further improved power by including in the trans-eQTL mapping model not only the etrait covariate for regulator gene *j* associated with the trans-eQTL, but also the effect of a cis-eQTL affecting target gene *i*.

Genetic Markers for eQTL Mapping

In a Genetical Genomics experiment, a segregating population of hundreds of individuals must be both expression profiled and marker genotyped on a genome-wide scale. Both types of large-scale profiling are expensive and time consuming. It was therefore a major breakthrough when it was realized that the expression data obtained by using Affymetrix chips could be used for *in silico* genome-wide marker genotyping by identifying so-called Single Feature Polymorphisms (SFP). Several groups, including our own, have now used SFP markers for eQTL mapping and the current evidence suggests that SFPs are reliable, provide quite dense coverage of the genome, and integrate well with existing conventional marker maps (Borevitz et al., 2003; Cui et al., 2005; Luo et al., 2007; Ronald, Brem, Whittle & Kruglyak, 2005; Rostoks et al., 2005a; Rostoks et al., 2005b; West et al., 2007), at least when using mapping populations with only two genotypes per polymorphism, in particular RIL populations. The quality of the SFP genotype data depends on the quality of the *in silico* SFP discovery and genotyping algorithm, and several such algorithms have been suggested at the present time (see (Luo et al., 2007) for a comparison among several methods). At this time, SFP typing has been mostly performed in populations with only two possible genotypes at each locus (in particular in RIL populations). How well SFP typing will work in other populations with three or more genotypes (e.g. intercrosses, human populations) is not yet known. Further validations of this SFP genotyping methodology and applications to other mapping populations are expected in the near future.

SELECTION OF REGULATOR – TARGET PAIRS

The selection of candidate regulators or regulator-target pairs for each identified eQTL described in this section depends on the availability of sequence information, i.e. knowledge of the genomic location of the expression profiled genes relative to the markers and eQTL regions. The outcome of the selections presented here is a strongly constrained GRN space.

The problem of identifying candidate regulatory genes from eQTL confidence regions has been approached with various methods. Some authors consider one eQTL at a time to select candidate regulators (Bing & Hoeschele, 2005; Keurentjes et al., 2007), while others simultaneously consider all eQTLs affecting a given etrait (Li et al., 2005). Bing and Hoeschele used partial correlation tests to identify candidate regulator genes located in the identified eQTL regions. In this approach, correlations between the etraits of genes located in the eQTL and the etraits affected by the eQTL are evaluated, since the etrait of the candidate regulator gene containing the causal polymorphism underlying the eQTL should correlate with the etrait(s) of the target gene(s) of the eQTL most strongly. But correlations can be due to indirect rather than direct causal influences or due to confounding (de la Fuente, Bing, Hoeschele

& Mendes, 2004; Shipley, 2002). To identify only the direct causal influences, partial correlations are calculated between the etrait of any gene located in an eQTL and the etrait of any target gene affected by the eQTL, conditional on the etraits of one (first order partial correlation) or two (second order partial correlation) other genes that are also located in the eQTL.

Li *et al*. (Li et al., 2005) use analysis of between-strain Single Nucleotide polymorphisms (SNPs) to exclude many possible candidate genes. Data for around 3 million SNPs are available for the two progenitor strains used in their study. These dense SNP data were used to determine whether the coding regions of the candidate regulator genes are identical by descent in the parents. Only genes with missense and nonsense SNPs were considered as potential regulators (Li et al., 2005).

The accuracy of the candidate regulator selection for each eQTL is clearly limited when using Genetical Genomics data alone. Therefore, it is very important to incorporate additional (external) biological information such as the SNP data, and several other approaches have been proposed. Keurentjes *et al*. (Keurentjes et al., 2007) first ranked the candidate regulators based on correlations and then selected candidates using the Iterative Group Analysis approach (Breitling, Amtmann & Herzyk, 2004). Stylianou et al. use automated literature database and manual search to find candidate genes (Stylianou et al., 2006). Tu et al. use a stochastic algorithm to also incorporate available protein-protein interaction, protein phosphorylation, and transcription factor–DNA binding information (Tu, Wang, Arbeitman, Chen & Sun, 2006).

Lee et al. proposed a probabilistic method called "Geronemo", which extends the module network approach of Segal et al. (Segal et al., 2003) to incorporate both expression and marker genotype data (Lee, Pe'er, Dudley, Church & Koller, 2006). Their approach iterates between the following steps until convergence is reached: 1) Assign genes to regulatory modules with clustering. 2) Learn the network for each module using a Bayesian scoring approach. With this approach, they were able to detect regulatory relationships that are indiscernible when genes are considered in isolation (Lee, Pe'er, Dudley, Church & Koller, 2006).

Liu et al. (Liu, de la Fuente & Hoeschele, 2008) use local regression models separately for each eQTL to identify regulator-target pairs, taking into account that the candidate regulators may affect a target through cis, trans or cis-trans regulation. Given the results from cis, cistrans and trans eQTL mapping and from other non-sequence based QTL mapping methods, regulator-target pairs were selected in several steps: 1) For each identified cis-eQTL affecting several potential cis-regulated genes (these genes are all affected by the same marker or eQTL and co-locate with it), for each potentially cis-regulated gene it is determined whether the gene is most likely truly cis-regulated or more likely cistrans affected. 2) For any eQTL where gene *t* was identified as a target (affected by but not co-located with the eQTL) and gene *r* was identified as cis-regulated (affected by and co-located with the eQTL), it is determined whether *r* is most likely trans or cistrans regulated. 3) For each target genes *t* retained in step 2) in an eQTL region, the most likely candidate regulator gene *r* (located in the eQTL) is determined. The determinations in steps 1) and 2) can be made based on the regression model

$$y_{tn} = \mu + b_1 y_{rn} + b_2 x_n + \varepsilon_{tn}; \qquad n = 1,\ldots,N \tag{2}$$

where y_t is the etrait value of target gene, μ is the overall mean of etrait *t*, y_r is the etrait value of regulator gene, and *x* is genotype indicator of the eQTL (marker). In steps 1) and 2), if the null hypothesis $b_2 = 0$ cannot be rejected for some gene r, then a cistrans regulation of *t* is indicated. For step 3), an additional

term ($b_3 y_{r',n}$) for another candidate regulator r' in the same eQTL is added to model (Eq. 2), and *r* is retained as candidate regulator of *t* only if the null hypothesis $b_1 = 0$ is rejected in the presence of any $r' \neq r$. 4) Lastly, using the results from trans-eQTL mapping, for each target gene *t* with at least two identified regulator genes, for each identified regulator *r* of *t*, another identified regulator (*r'*) of *t* and its nearest marker are included in model (Eq. 2) to check whether the null hypothesis $b_1 = 0$ can be rejected in the presence of any other regulator $r' \neq r$, in which case *r* is retained as a candidate regulator of *t*..

GRN INFERENCE IN THE CONSTRAINED NETWORK SPACE

In this section we review approaches for search and evaluation of global network models within the strongly constrained network model space as defined by the eQTL analysis and candidate regulator and regulator-target pairs selection. Two main approaches have been taken. The first approach uses the eQTL mapping results to define constraints on the network space, and the search is conducted using Bayesian Network analysis (Zhu et al., 2004). The second approach first constructs an Encompassing Directed Network (EDN) based on the results from eQTL mapping and regulator-target pair selection by assembling all retained candidate regulator target pairs and then searches for an optimal network model embedded in this EDN using Structural Equation Modeling (Liu, de la Fuente & Hoeschele, 2008).

The first approach makes use of the fact that it is possible to derive constraints on the GRN space and to perform causal inference in the absence of sequence information (and other external biological information). This analysis will have (maybe substantially) reduced power, but it is relevant not only for organisms where sequence information is not yet available, but also when the quantities of interest are not (just) the gene-expressions, but also or instead include phenotypic traits, metabolomic profiles etc.. In these cases we cannot establish causality based on a regulator (gene) being located in an eQTL that affects a target (gene). To infer a gene regulatory (causal) network without sequence information, and based on the results from the eQTL analysis, Zhu et al. (Zhu et al., 2004) consider regulations only among any two genes whose etraits have common eQTLs. For any two genes whose etraits do not share at least one eQTL, it is assumed that no regulatory relationship exists. To quantify the extent of 'QTL overlap', a weighted average correlation was used. The motivation behind this constraint is that if two etraits are controlled by the same eQTL, then either they are independently affected by the eQTL, or etrait 1 is directly affected by the eQTL and in turn affects etrait 2, or etrait 2 is directly affected by the eQTL and in turn affects etrait 1. Schadt *et al*. (Schadt et al., 2005) use a Likelihood-based Causality Model Selection (LCMS) method to select the most likely one of these three cases based on the Akaike information criterion. The eQTL overlap of two etraits does not need to be complete as any gene and its etrait can have several inputs associated with different eQTLs. In fact (not noted by those authors), it is helpful for the etraits of two genes not to share all their eQTLs. For example, suppose that two genes (etraits) 1 and 2 share a subset of eQTLs (subset A), while another subset of eQTLs (subset B) affects only etrait 2. Then, the evidence for regulation of gene B by gene A, as opposed to vice versa, is stronger than it would be without marker(s) B. Several other papers deal with the detection of candidate genes by using overlap of eQTLs and complex trait QTLs, including (Chen et al., 2008; Chesler et al., 2005; Cheung et al., 2005; DeCook, Lall, Nettleton & Howell, 2006; Schadt et al., 2003).

Bayesian Networks

The Bayesian Network (BN) approach has been applied to GRN inference from gene-expression data soon after the first datasets appeared (Friedman, 2004; Friedman, Linial, Nachman & Pe'er, 2000; Murphy & Mian, 1999). Using gene-expression data alone and without any constraints on the network space, causal inference for relationships among genes is very limited (see the book by Shipley (Shipley, 2002) on how to direct some edges in an undirected network derived from observational data or without interventions), and BN analysis is computationally very demanding and becomes infeasible for hundreds (or even thousands) of genes. Zhu et al. (Zhu et al., 2004) proposed to infer GRNs with a BN approach and a local structure search algorithm after constraining the network (structure) space by using common eQTLs as described in the previous section and by limiting the number of regulators per gene to at most three. Recently, these authors showed with a simulation study that GRN inference from Genetical Genomics data using BNs was much more accurate than GRN inference from expression data alone (Zhu et al., 2007), a very expected finding. Other authors including (Li et al., 2005) also employed a BN approach in a constrained search space defined by the eQTL mapping results.

BNs can be graphically represented as Directed Acyclic Graphs (DAG), *i.e.* networks in which no directed cycles are present (Pearl, 2000; Spirtes, Glymour & Scheines, 1993). The graphical model represents a conditional distribution for each node given its parents. The full joint distribution is defined as the product of the local conditional distributions. For BNs, the global directed Markov property permits the joint probability distribution of the variables to be factored according to the DAG (Pearl, 2000; Spirtes, Glymour & Scheines, 1993). For this reason the assumption of an acyclic network is so attractive: The factorization implies that only local likelihoods need to be calculated, which is computationally much more efficient than evaluating joint likelihoods involving possibly many variables. Let V be the random variable associated with a particular node (an etrait in our context). The factorization can be represented as, $p\left(V_1, V_2, \ldots, V_n\right) = \prod_{j=1}^{n} p\left(V_j \mid V\left(\text{parents of } j\right), , _j\right)$ where V (parents of j) is a vector of V's of the parent vertices of vertex j, and θ_j is the parameter vector of the local likelihood p(V_j |.) (Pearl, 2000). Therefore, the likelihood for each target etrait can be maximized separately. This factorization is a major computational simplification.

Model evaluation with BNs includes fitting parameters of each conditional probability distribution and search for the network structure (the graph topology). Structure learning is in general an NP-hard problem (Chickering, 2002), and many (heuristic) search algorithms are available, including greedy hill-climbing, greedy search with restarts, simulated annealing, and Monte-Carlo methods. For a comprehensive introduction to BNs, we refer the reader to the book by Jensen and Nielsen (Jensen & Nielsen, 2007), and specifically on 'learning' with BNs, to Heckerman's book chapter (Heckerman, 1999).

Being defined as DAGs, BNs cannot represent networks with cyclic relationships. However, there is strong evidence for GRNs to contain directed cycles (A. de la Fuente – this book). Recently, Chen et al. obtained evidence for extensive feedback control in the network they studied, due to the fact that strongly perturbing some genes in the network induced significant expression changes in a large number of the genes in the network (Chen et al., 2008). GRNs are therefore better modeled as Directed Cyclic Graphs (DCGs) (Liu, de la Fuente & Hoeschele, 2008) (see also A. de la Fuente – this book). Based on the assumption that a cyclic graph represents a dynamic system at equilibrium (Fisher, 1970), this problem can be theoretically resolved by including a time dimension, which produces causal graphs without cycles (DAGs) that can then be studied using BNs, an approach called Dynamic BNs (Hartemink, Gifford,

Jaakkola & Young, 2002; Murphy & Mian, 1999). However, this approach requires the collection of time series data, which is difficult to accomplish, as it requires synchronization of cells and close time intervals not allowing for feedback (Spirtes et al., 2000). Samples at wider time intervals represent near steady state data and hence require cyclic network reconstruction.

Structural Equation Modeling and Network Search

Structural Equation Modeling **(**SEM) is a linear statistical modeling framework that has been widely used in econometrics, sociology and psychology, usually as a confirmatory procedure instead of an exploratory analysis for causal inference (Bollen, 1989; Johnston, 1972; Judge, Griffiths, R.C, Ltkepohl & Lee, 1985). Shipley (Shipley, 2002) discusses the use of SEM in biology with an emphasis on causal inference. SEM has been used for association and linkage mapping of QTL (e.g.Neale, Boker, Xie & Maes, 2003; Stein et al., 2003). Xiong *et al*. (Xiong, Li & Fang, 2004) were the first to apply SEM to GRN reconstruction using gene-expression data (outside of the Genetical Genomics context). Their application was limited to GRNs without cyclic relationships by using a recursive SEM, which has an acyclic structure and uncorrelated errors and is equivalent to a Gaussian BN. These authors reconstructed only small networks with less than 20 genes. Li *et al*. (Li et al., 2006a) analyzed both phenotypic and DNA marker data on a segregating population to construct networks including a small number sub-phenotypes and QTL related to obesity and bone geometry, by SEM analysis using standard SEM software.

In the context of Genetical Genomics experiments, Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) developed an SEM analysis for GRN inference within the constrained network search space obtained from the eQTL mapping results and the regulator-target pair selection, which produced three lists of causal regulatory relationships: (1) a list containing all identified cis-regulations (eQTL A affects gene A located in its confidence region), (2) a list containing all cistrans regulations (cis-regulated gene A regulates gene B), and (3) a list containing all trans regulations (gene A regulates gene B and eQTL A affects gene B (but not gene A)). All the identified and retained regulator-target relationships were assembled into the EDN, which consisted of directed edges from eQTLs to cis-regulated target genes, from cis-regulated genes to cistrans regulated target genes, from trans-regulator genes to target genes and from trans-eQTLs to target genes. The EDN consisted of two types of nodes: Continuous nodes for the genes (etraits), and discrete nodes for the eQTLs (genotypes). The EDN thus defines a constrained network search space as the GRN we wish to identify is embedded in the EDN. Additional constraints were considered: certain edges cannot be removed from the EDN, because their removal would contradict the results from the eQTL analysis. If an etrait was found to be influenced by an eQTL, then there must remain either a direct or indirect path from the eQTL to that etrait's gene in the network. Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) then employed SEM to evaluate models within this model space. Due to the fact that the EDN contained many cycles, BN approaches could not be used. In contrast, SEM can be applied to cyclic network inference.

In general, SEM consists of a structural model describing (causal) relationships among latent variables and a measurement model describing the relationships between the observed measurements and the underlying latent variables. Any SEM can be represented both algebraically as well as graphically. Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) use SEM with observed variables only (there is no measurement model), which can be represented as

$$\boldsymbol{y}_i = \boldsymbol{B}\boldsymbol{y}_i + \boldsymbol{F}\boldsymbol{x}_i + \boldsymbol{e}_i; \quad \boldsymbol{e}_i \sim (\boldsymbol{0}, \boldsymbol{E}) \qquad i = 1, ..., \mathrm{N} \tag{3}$$

In this model, for member i of the segregating population ($i = 1, \ldots, N$), $\boldsymbol{y}_i = (y_{i1},...,y_{ip})^{\mathrm{T}}$ is the vector of expression values of all (p) genes in the network, and $\boldsymbol{x}_i = (x_{i1},...,x_{iq})^{\mathrm{T}}$ denotes the vector of marker or eQTL genotype codes. The $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ are deviations from means, $\boldsymbol{e}_i$ is a vector of error terms, and $\boldsymbol{E}$ is its covariance matrix (we note that a mean structure could be incorporated in the SEM as needed). Matrix $\boldsymbol{B}$ contains coefficients for the direct causal effects of the etraits on each other. Matrix $\boldsymbol{F}$ contains coefficients for the direct causal effects of the eQTLs on the etraits. The structure of matrices $\boldsymbol{B}$ and $\boldsymbol{F}$ corresponds to the path diagram or directed graph representing the structural model, in which vertices or nodes represent genes and eQTLs, and edges correspond to the non-zero elements in $\boldsymbol{B}$ and $\boldsymbol{F}$. Matrices $\boldsymbol{B}$ and $\boldsymbol{F}$ are sparse when the model represents a sparse network. When the elements in $\boldsymbol{e}_i$ are uncorrelated and matrix $\boldsymbol{B}$ can be rearranged as a lower triangular matrix, the model is recursive, there are no cyclic relationships, and the corresponding graph is a Directed Acyclic Graph (DAG). If the error terms are correlated ($\boldsymbol{E}$ is not diagonal), or matrix $\boldsymbol{B}$ cannot be rearranged into a triangular matrix (indicating the presence of cycles), the model is non-recursive. The graph corresponding to a non-triangular matrix $\boldsymbol{B}$ is a Directed Cyclic Graph (DCG).

In Genetical Genomics experiments, the $\boldsymbol{x}_i$ are random vectors because individuals are sampled from a segregating population. However, the joint likelihood of the $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ can be factored into the conditional likelihood of the $\boldsymbol{y}_i$ given the $\boldsymbol{x}_i$ times the likelihood of the $\boldsymbol{x}_i$, and the latter does not depend on any of the network parameters in $\boldsymbol{B}$, $\boldsymbol{F}$ and $\boldsymbol{E}$ and can therefore be ignored. Thus, we only need to assume multivariate normality for the residual vectors $\boldsymbol{e}_i$ when specifying the likelihood function.

Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) factor the joint likelihood function of the $\{\boldsymbol{y}_i ; \mathrm{i}=1,\ldots,\mathrm{N}\}$ into a product of local likelihoods which depend on different sets of parameters and are maximized individually in analogy with BN analysis. A network with cyclic components (systems of connected cycles, in which any gene can find a path back to itself through other genes) becomes acyclic when a set of genes pertaining to the same cyclic component is collapsed into a single node. The joint likelihood can therefore be factored as a product of conditional likelihoods pertaining to individual genes which do not belong to any cyclic component, and of conditional joint likelihoods each pertaining to a set of genes in a cyclic component. For the genes involved in a cyclic component, their joint likelihood was maximized using a Genetic Algorithm (GA) based global optimization procedure. The constrained network space defined by the EDN will typically be still much too large to exhaustively enumerate all possible network structures. Therefore, a heuristic search strategy based on the principle of Occam's Window model selection (Madigan & Raftery, 1994), which potentially selects multiple acceptable models, was adapted. Alternative models or network structures were compared using the Bayesian Information Criteron (Raftery, 1993). The selection of multiple models may be important for two reasons: First, the data may provide (nearly) the same support for multiple models, and this information would otherwise be missed. Secondly, for DCGs it can happen that two models with different edges have the same likelihood (they are equivalent) (Richardson, 1996; Richardson & Spirtes, 1999). In contrast, two DAG models can only be equivalent if they have the same edges but differ in the direction of an edge (Pearl, 2000), and this equivalence would not occur with Genetical Genomics data where the edge directions are fixed by the eQTL information. Based on the factorization of the overall joint likelihood, the strongly constrained network topology search space defined by the EDN, and a careful choice of starting values for GA optimization, the algorithm proposed by Liu *et al*. (Liu, de la Fuente & Hoeschele, 2008) can

infer GRNs of hundreds of variables. In (Mancosu et al., 2008) a computationally very efficient local 'sparsification' approach rather than using the global model selection approach via SEM. This simplification made it feasible to analyze a genome scale yeast dataset, thus compiling a genome-wide yeast GRN (see figure 1 in A. de la Fuente – this book). It is important however to evaluate the final model(s) selected via SEM in order to verify that the models fit the data sufficiently well.

Undirected Networks in the Context of eQTLs

Multiple studies have inferred Co-Expression Networks (see A. de la Fuente – this book) using association measures between the etraits of pairs of genes (Butte et al. 2000; Wille et al. 2003; Magwene and Kim 2004), or partial correlation resulting in an approximate Undirected Dependency Graph (UDG) (de la Fuente, Bing, Hoeschele & Mendes, 2004; Shipley, 2002). We note that some other authors have used Graphical Gaussian Modeling (or covariance selection) (e.g.Schäfer & Strimmer, 2005a,, 2005b), but in this approach of constructing an undirected graph there is an edge between any two genes whose partial correlation conditional on all other genes has been found to be nonzero. As opposed to the UDG, such a graph would contain an edge between two genes that do not regulate each other and are not regulated by some common cause but jointly regulate another ('child') gene. In any of these graphs, an edge is retained if the corresponding correlation coefficient exceeds a chosen threshold or has been found to be significant by a statistical correlation test. Consequently, between any two genes an edge either exists or does not exist. As an alternative, weighted Co-Expression Networks have been proposed, where first a matrix of the absolute values of the simple correlations between any two genes is computed, which is then converted into a matrix of 'connection strengths' using a power function of the absolute correlation coefficients (Zhang & Horvath, 2005). This weighted network is seen as being robust, in contrast with the other (unweighted) undirected networks which depend on a chosen (significance) threshold.

Some studies have combined these undirected networks with eQTL information. An undirected network can be constructed in the context of Genetical Genomics by only using the expression data. Genotype data can help to reduce the number of false positive edges. For example, when constructing co-expression networks, Lum et al. required a pair of linked etrait nodes to be regulated by at least one common eQTL (Lum et al., 2006). Ghazalpour *et al*. (Ghazalpour et al., 2006) constructed weighted Co-Expression Networks and identified highly interconnected network modules. They then detected module-specific "genomic hotspots" (mQTLs) that regulate the expression of these modules, and they investigated the co-location of these mQTL with physiological traits of mice. With mouse data, Chen *et al*. (Chen et al., 2008) constructed Co-Expression Networks using both genotype and expression data as in Lum et al. (Lum et al., 2006), and detected highly interconnected modules in the constructed co-expression networks using an iterative search algorithm. They then established directed relationships between the QTLs, metabolic traits and etraits using the LCMS method as described previously (Schadt et al., 2005). A sub-network was detected as having a causal relationship with the metabolic traits of interest if the sub-network was enriched for etraits that had causal associations with the metabolic traits. Emilsson *et al*. (Emilsson et al., 2008) constructed co-expression networks with similar approaches using human data, and identified a sub-network that was highly conserved in mice – the macrophage-enriched network (Chen et al., 2008). They performed cis-eQTL mapping for this network and found that the cis-eQTLs detected showed some evidence of association to obesity related traits (Emilsson et al., 2008).

It is possible to obtain a regulatory or causal network in the absence of sequence information (when an organism does not yet have a sequence assembly, or when working with non-expression phenotypes

such as metabolic traits) by first constructing an undirected network using only the phenotypes, then performing QTL mapping, and subsequently orienting all edges in the network for which there is QTL information by using and further developing an approach such as the LCMS method, *i.e.* by using local structural equation modeling. Instead of a multi-step approach, where an undirected network and highly connected modules are first identified and then module-QTLs are detected, one may search directly for QTLs, or groups of QTLs, that jointly regulate groups of genes, e.g. by adapting CCA as mentioned earlier.

CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter, we have reviewed Gene Regulatory Network inference with 'Genetical Genomics' data (Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002; Jansen, 2003; Jansen & Nap, 2001; Schadt et al., 2003). For a detailed biological application of similar approaches, please refer to another chapter in this book "Gene-expression Regulatory Regions in Yeast Amino Acid Biosynthetic Pathways Unveiled by Quantitative Trait Locus Mapping". An important component of any Genetical Genomics analysis is the mapping of eQTL. Although eQTL mapping was initially performed by simply using standard QTL mapping methods and software (developed for the analysis of one or few phenotypic traits), it was soon recognized that such analyses are very sub-optimal and miss important information. Clearly the quality of GRN inference crucially depends on the quality of the eQTL mapping results. There is still a critical need for further development and evaluation of methodology and software for eQTL mapping, such as developing full Bayesian analyses modeling individual eQTL – etrait relationships or involving dimension reduction such as Canonical Correlation Analysis, developing and using more appropriate criteria and controlling methods for multiple testing, and identifying epistatic eQTL.

Likewise, further development and evaluation of methods for GRN inference in the constrained network search space is still warranted. We previously have proposed SEM due to its ability to fit cyclic network models (Liu, de la Fuente & Hoeschele, 2008). Our current implementation of SEM, which is capable of analyzing networks with a few hundred gene and eQTL nodes, uses a heuristic search strategy and maximum likelihood inference. A Markov Chain Monte Carlo Bayesian implementation of SEM would have multiple advantages, including an ability to incorporate prior information, an ability to select multiple models and represent uncertainty about the network model (and the values of its parameters) given the data, and possibly an ability to analyze a larger network space. The use of the Bayesian Information criterion and related criteria for network model selection and the use of sparsity priors in a Bayesian analysis would strongly favor sparse networks, although bio-molecular systems are not necessarily most parsimonious. For at least some of the edges (regulator-target pairs) in the EDN, there may be prior biological knowledge from various sources, for example transcription factor binding location data, information on pathway relationships (Franke et al., 2006), SNP presence in candidate regulators (Li et al., 2005), and information on protein-protein interactions (Tu, Wang, Arbeitman, Chen & Sun, 2006). A principled incorporation of such prior knowledge into methods for GRN reconstruction from microarray data has been considered by several authors via prior distributions in Bayesian analysis (e.g.Bernard & Hartemink, 2005; Imoto et al., 2002; e.g.Werhli & Husmeier, 2007).

As Genetical Genomics studies typically involve a segregating population with at least near one hundred or several hundreds of individuals, there is a large expense for genome-wide expression profiling of all individuals and, when not relying on *in silico* SFP typing, then there is a similarly large

expense for genome-wide DNA marker typing. It is therefore important to develop analysis methods that extract the most information from the data, as discussed above. There are also considerations related to optimal study design given limited resources. For two color microarrays, Fu and Jansen (Fu & Jansen, 2006) proposed a 'distant pair design' to maximize genetic dissimilarity between individuals on the same array to maximize power for decomposing expression variation. Because genotyping is in general expensive, some studies used 'selective genotyping', *i.e.* all individuals are phenotyped while only selected individuals are genotyped (e.g.Jannink, 2005; e.g.Medugorac & Soller, 2001). Selective genotyping is well-established in the QTL literature (e.g.Lander & Botstein, 1989); by genotyping only those individuals whose phenotypes are extreme (in the tails of the distribution of phenotypes), the same amount of information is obtained as when genotyping a larger number of individuals randomly. This approach works well for a single phenotype while it is difficult or not useful in the context of multiple phenotypes. In Genetical Genomics, there are at least several thousands of expression phenotypes, and expression profiling may be more expensive than marker typing. Therefore, selective expression profiling approaches have been studied (Bueno Filho, Gilmour & Rosa, 2006; Jin et al., 2004; Wang & Nettleton, 2006). There is a need for algorithms that search for optimal designs. For further review of design issues in Genetical Genomics experiments, the reader should consult (Kendziorski & Wang, 2006; Rosa, de Leon & Rosa, 2006).

This chapter focused on GRN inference in the context of Genetical Genomics studies (Rockman, 2008). It focused on GRN inference for a single organism. An important and necessary extension is to infer the genetic interactome of multiple organisms in host-pathogen interaction studies, where both the host and the pathogen are expression profiled (Zhou et al., 2008). Beyond expression profiling and GRN inference, Systems Genetics (Threadgill, 2006) will allow us to infer integrated causal networks including other molecular phenotypes, such as proteomics data (e.g.Foss et al., 2007; Peck, 2005), metabolomics data (Keurentjes et al., 2006), and organismal phenotypes (Li et al., 2006a; Nadeau et al., 2003). This will require the sequence-based and not-sequence based causal inference algorithms using eQTL information, as described above, to be more fully developed and combined.

GRN reverse-engineers have relied on very expensive and difficult to perform single-gene perturbation experiments and time series experiments, and they are still eagerly awaiting the appearance of datasets with a large number of experimental observations. Fortunately, such datasets are currently appearing using a Genetic Genomics (or Systems Genetics) setup, in which genotyping and gene-expression profiling are performed on a genetically randomized population of individuals. Like (artificial) single gene perturbations, genetic segregation at many loci can be used to establish causal relationships between genes (Jansen, 2003; Jansen & Nap, 2001). Several such datasets are available for yeast (Brem & Kruglyak, 2005; Brem, Yvert, Clinton & Kruglyak, 2002), *Arabidopsis* (Keurentjes et al., 2007; Vuylsteke, van Eeuwijk, Van Hummelen, Kuiper & Zabeau, 2005; West et al., 2007), *C. elegans* (Li et al., 2006b), fruit fly (Anholt et al., 2003), mouse (Bystrykh et al., 2005; Schadt et al., 2003), soybean (Zhou et al., 2008), and human (Cheung et al., 2003; Cheung et al., 2005; Goring et al., 2007), with sample sizes ranging from near one hundred to more than a thousand of observations. Genetical Genomics datasets with large sample sizes are relatively cheap to produce as compared to artificial single gene perturbations. As pointed out earlier, the multi-factorial and 'natural' properties of the Genetical Genomics perturbations have clear advantages over the mostly single gene (Hughes et al., 2000; Mnaimneh et al., 2004) or pairs of genes (Tong et al., 2004) artificial perturbations. We therefore expect Genetical Genomics and Systems Genetics to be a major source of data for inferring Gene Regulatory Networks and more general causal Bio-Molecular Networks in the near future.

REFERENCES

Anholt, R. R., Dilda, C. L., Chang, S., Fanara, J. J., Kulkarni, N. H., & Ganguly, I. (2003). The genetic architecture of odor-guided behavior in Drosophila: Epistasis and the transcriptome. *Nature Genetics, 35*(2), 180–184. doi:10.1038/ng1240

Barabasi, A. L., & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews. Genetics*, *5*(2), 101–113. doi:10.1038/nrg1272

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate—a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B. Methodological*, *57*, 289–300.

Bernard, A., & Hartemink, A. J. (2005). Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 459–470.

Beyene, J., Tritchler, D., Bull, S. B., Cartier, K. C., Jonasdottir, G., & Kraja, A. T. (2007). Multivariate analysis of complex gene expression and clinical phenotypes with genetic marker data. *Genetic Epidemiology*, *31*(Suppl 1), S103–S109. doi:10.1002/gepi.20286

Bing, N., & Hoeschele, I. (2005). Genetical genomics analysis of a yeast segregant population for transcription network inference. *Genetics*, *170*(2), 533–542. doi:10.1534/genetics.105.041103

Bollen, K. (1989). *Structural equations with latent variable.* Wiley-Interscience.

Boomsma, D. (1996). Using multivariate genetic modeling to detect pleiotropic quantitative trait loci. *Behavior Genetics*, *26*(2), 161–166. doi:10.1007/BF02359893

Borevitz, J. O., Liang, D., Plouffe, D., Chang, H. S., Zhu, T., & Weigel, D. (2003). Large-scale identification of single-feature polymorphisms in complex genomes. *Genome Research*, *13*(3), 513–523. doi:10.1101/gr.541303

Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene networks: How to put the function in genomics. *Trends in Biotechnology*, *20*(11), 467–472. doi:10.1016/S0167-7799(02)02053-X

Breitling, R., Amtmann, A., & Herzyk, P. (2004). Iterative group analysis (iGA): A simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, *5*, 34. doi:10.1186/1471-2105-5-34

Brem, R. B., & Kruglyak, L. (2005). The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(5), 1572–1577. doi:10.1073/pnas.0408709102

Brem, R. B., Storey, J. D., Whittle, J., & Kruglyak, L. (2005). Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature*, *436*(7051), 701–703. doi:10.1038/nature03865

Brem, R. B., Yvert, G., Clinton, R., & Kruglyak, L. (2002). Genetic dissection of transcriptional regulation in budding yeast. *Science*, *296*(5568), 752–755. doi:10.1126/science.1069516

Bueno Filho, J. S. S., Gilmour, S. G., & Rosa, G. J. M. (2006). Design of microarray experiments for genetical genomics studies. *Genetics*, *174*(2), 945–957. doi:10.1534/genetics.106.057281

Bystrykh, L., Weersing, E., Dontje, B., Sutton, S., Pletcher, M. T., & Wiltshire, T. (2005). Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics'. *Nature Genetics*, *37*(3), 225–232. doi:10.1038/ng1497

Carlborg, O., De Koning, D. J., Manly, K. F., Chesler, E., Williams, R. W., & Haley, C. S. (2005). Methodological aspects of the genetic dissection of gene expression. *Bioinformatics (Oxford, England)*, *21*(10), 2383–2393. doi:10.1093/bioinformatics/bti241

Casella, G., & Berger, R. L. (1990). *Statistical inference*. Pacific Grove, CA: Wadsworth.

Chen, L., & Storey, J. D. (2006). Relaxed significance criteria for linkage analysis. *Genetics*, *173*, 2371–2381. doi:10.1534/genetics.105.052506

Chen, Y., Zhu, J., Lum, P. Y., Yang, X., Pinto, S., Macneil, D. J., et al. (2008). Variations in DNA elucidate molecular networks that cause disease. *Nature*.

Chesler, E. J., Lu, L., Shou, S., Qu, Y., Gu, J., & Wang, J. (2005). Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, *37*(3), 233–242. doi:10.1038/ng1518

Cheung, V. G., Jen, K. Y., Weber, T., Morley, M., Devlin, J. L., & Ewens, K. G. (2003). Genetics of quantitative variation in human gene expression. *Cold Spring Harbor Symposia on Quantitative Biology*, *68*, 403–407.

Cheung, V. G., Spielman, R. S., Ewens, K. G., Weber, T. M., Morley, M., & Burdick, J. T. (2005). Mapping determinants of human gene expression by regional and genome-wide association. *Nature*, *437*(7063), 1365–1369. doi:10.1038/nature04244

Chickering, D. M. (2002). Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, *2*, 445–498. doi:10.1162/153244302760200696

Churchill, G. A., & Doerge, R. W. (1994). Empirical threshold values for quantitative trait mapping. *Genetics*, *138*(3), 963–971.

Comuzzie, A. G., Mahaney, M. C., Almasy, L., Dyer, T. D., & Blangero, J. (1997). Exploiting pleiotropy to map genes for oligogenic phenotypes using extended pedigree data. *Genetic Epidemiology*, *14*(6), 975–980. doi:10.1002/(SICI)1098-2272(1997)14:6<975::AID-GEPI69>3.0.CO;2-I

Cui, X., Xu, J., Asghar, R., Condamine, P., Svensson, J. T., & Wanamaker, S. (2005). Detecting single-feature polymorphisms using oligonucleotide arrays and robustified projection pursuit. *Bioinformatics (Oxford, England)*, *21*(20), 3852–3858. doi:10.1093/bioinformatics/bti640

D'Haeseleer, P., Liang, S., & Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics (Oxford, England)*, *16*(8), 707–726. doi:10.1093/bioinformatics/16.8.707

Darvasi, A. (1998). Experimental strategies for the genetic dissection of complex traits in animal models. *Nature Genetics*, *18*(1), 19–24. doi:10.1038/ng0198-19

de la Fuente, A., Bing, N., Hoeschele, I., & Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics (Oxford, England)*, *20*, 3565–3574. doi:10.1093/bioinformatics/bth445

de la Fuente, A., Brazhnik, P., & Mendes, P. (2002). Linking the genes: Inferring quantitative gene networks from microarray data. *Trends in Genetics*, *18*(8), 395–398. doi:10.1016/S0168-9525(02)02692-6

DeCook, R., Lall, S., Nettleton, D., & Howell, S. H. (2006). Genetic regulation of gene expression during shoot development in Arabidopsis. *Genetics*, *172*(2), 1155–1164. doi:10.1534/genetics.105.042275

di Bernardo, D., Thompson, M. J., Gardner, T. S., Chobot, S. E., Eastwood, E. L., & Wojtovich, A. P. (2005). Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nature Biotechnology*, *23*(3), 377–383. doi:10.1038/nbt1075

Doerge, R. W. (2002). Mapping and analysis of quantitative trait loci in experimental populations. *Nature Reviews. Genetics*, *3*(1), 43–52. doi:10.1038/nrg703

Doerge, R. W., Zeng, Z.-B., & Weir, B. S. (1997). Statistical issues in the search for genes affecting quantitative traits in experimental populations. *Statistical Science*, *12*, 195–219. doi:10.1214/ss/1030037909

Dorogovtsev, S. N., & Mendes, J. F. F. (2003). *Evolution of networks: From biological networks to the Internet and WWW*. Oxford: Oxford University Press.

Doss, S., Schadt, E. E., Drake, T. A., & Lusis, A. J. (2005). Cis-acting expression quantitative trait loci in mice. *Genome Research*, *15*(5), 681–691. doi:10.1101/gr.3216905

Emilsson, V., Thorleifsson, G., Zhang, B., Leonardson, A. S., Zink, F., Zhu, J., et al. (2008). Genetics of gene expression and its effect on disease. *Nature*.

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., & Cottarel, G. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, *5*(1), e8. doi:10.1371/journal.pbio.0050008

Fisher, F. M. (1970). A correspondence principle for simultanious equation models. *Econometrica*, *38*(1), 73–92. doi:10.2307/1909242

Fisher, R. A. (1954). *Statistical methods for research workers* (12th edition). Edinburgh, UK.

Foss, E. J., Radulovic, D., Shaffer, S. A., Ruderfer, D. M., Bedalov, A., & Goodlett, D. R. (2007). Genetic basis of proteome variation in yeast. *Nature Genetics*, *39*(11), 1369–1375. doi:10.1038/ng.2007.22

Franke, L., Bakel, H., Fokkens, L., de Jong, E. D., Egmont-Petersen, M., & Wijmenga, C. (2006). Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *American Journal of Human Genetics*, *78*(6), 1011–1025. doi:10.1086/504300

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*(5659), 799–805. doi:10.1126/science.1094068

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3-4), 601–620. doi:10.1089/106652700750050961

Fu, J., & Jansen, R. C. (2006). Optimal design and analysis of genetic studies on gene expression. *Genetics*, *172*(3), 1993–1999. doi:10.1534/genetics.105.047001

Gardner, T., di Bernardo, D., Lorenz, D., & Collins, J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *301*(5629), 102–105. doi:10.1126/science.1081900

Ghazalpour, A., Doss, S., Zhang, B., Wang, S., Plaisier, C., & Castellanos, R. (2006). Integrating genetic and network analysis to characterize genes related to mouse weight. *PLOS Genetics*, *2*(8), e130. doi:10.1371/journal.pgen.0020130

Goring, H. H., Curran, J. E., Johnson, M. P., Dyer, T. D., Charlesworth, J., & Cole, S. A. (2007). Discovery of expression QTLs using large-scale transcriptional profiling in human lymphocytes. *Nature Genetics*, *39*(10), 1208–1216. doi:10.1038/ng2119

Hartemink, A., Gifford, D., Jaakkola, T., & Young, R. (2002). *Combining location and expression data for principled discovery of genetic regulatory network models.* Paper presented at the Pac. Symp. Biocomput.

Heckerman, D. (1999). A tutorial on learning with Bayesian networks. In M. Jordan (Ed.), *Learning in graphical models*. Cambridge: MIT Press

Hubner, N., Wallace, C. A., Zimdahl, H., Petretto, E., Schulz, H., & Maciver, F. (2005). Integrated transcriptional profiling and linkage analysis for identification of genes underlying disease. *Nature Genetics*, *37*(3), 243–253. doi:10.1038/ng1522

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., & Armour, C. D. (2000). Functional discovery via a compendium of expression profiles. *Cell*, *102*(1), 109–126. doi:10.1016/S0092-8674(00)00015-5

Imoto, S., Sunyong, K., Goto, T., Aburatani, S., Tashiro, K., Kuhara, S., et al. (2002). *Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network.* Paper presented at the Proc. IEEE Comput. Soc. Bioinform. Conf.

Jannink, J. L. (2005). Selective phenotyping to accurately map quantitative trait loci. *Crop Science*, *45*, 901–908. doi:10.2135/cropsci2004.0278

Jansen, R. C. (1993). Interval mapping of multiple quantitative trait loci. *Genetics*, *135*(1), 205–211.

Jansen, R. C. (2003). Studying complex biological systems using multifactorial perturbation. *Nature Reviews. Genetics*, *4*, 145–151. doi:10.1038/nrg996

Jansen, R. C., & Nap, J. P. (2001). Genetical genomics: The added value from segregation. *Trends in Genetics*, *17*(7), 388–391. doi:10.1016/S0168-9525(01)02310-1

Jensen, F. V., & Nielsen, T. D. (2007). *Bayesian networks and decision graphs* (2nd edition).

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., & Barabasi, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, *407*(6804), 651–654. doi:10.1038/35036627

Jia, Z., & Xu, S. (2007). Mapping quantitative trait loci for expression abundance. *Genetics*, *176*(1), 611–623. doi:10.1534/genetics.106.065599

Jiang, C., & Zeng, Z. B. (1995). Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics*, *140*(3), 1111–1127.

Jin, C., Lan, H., Attie, A. D., Churchill, G. A., Bulutuglo, D., & Yandell, B. S. (2004). Selective phenotyping for increased efficiency in genetic mapping studies. *Genetics*, *168*(4), 2285–2293. doi:10.1534/genetics.104.027524

Johnston, J. (1972). *Econometric methods* (2nd edition). St. Louis: McGraw-Hill.

Judge, G. G., & Griffiths, W. E. R.C, H., Ltkepohl, H., & Lee, T. C. (1985). *The theory and practice of econometrics*. Wiley.

Kacser, H., & Burns, J. A. (1981). The molecular basis of dominance. *Genetics*, *97*(3-4), 639–666.

Kendziorski, C., & Wang, P. (2006). A review of statistical methods for expression quantitative trait loci mapping. *Mammalian Genome*, *17*(6), 509–517. doi:10.1007/s00335-005-0189-6

Kendziorski, C. M., Chen, M., Yuan, M., Lan, H., & Attie, A. D. (2006). Statistical methods for expression quantitative trait loci (eQTL) mapping. *Biometrics*, *62*(1), 19–27. doi:10.1111/j.1541-0420.2005.00437.x

Keurentjes, J. J., Fu, J., de Vos, C. H., Lommen, A., Hall, R. D., & Bino, R. J. (2006). The genetics of plant metabolism. *Nature Genetics*, *38*(7), 842–849. doi:10.1038/ng1815

Keurentjes, J. J., Fu, J., Terpstra, I. R., Garcia, J. M., van den Ackerveken, G., & Snoek, L. B. (2007). Regulatory network construction in Arabidopsis by using genome-wide gene expression quantitative trait loci. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(5), 1708–1713. doi:10.1073/pnas.0610429104

Kulp, D., & Jagalur, M. (2006). Causal inference of regulator-target pairs by gene mapping of expression phenotypes. *BMC Genomics*, *7*(1), 125. doi:10.1186/1471-2164-7-125

Lan, H., Stoehr, J. P., Nadler, S. T., Schueler, K. L., Yandell, B. S., & Attie, A. D. (2003). Dimension reduction for mapping mRNA abundance as quantitative traits. *Genetics*, *164*(4), 1607–1614.

Lander, E., & Botstein, D. (1989). Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics*, *121*, 185–199.

Lander, E. S., & Schork, N. J. (1994). Genetic dissection of complex traits. *Science*, *265*(5181), 2037–2048. doi:10.1126/science.8091226

Lee, S. I., Pe'er, D., Dudley, A. M., Church, G. M., & Koller, D. (2006). Identifying regulatory mechanisms using individual variation reveals key role for chromatin modification. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(38), 14062–14067. doi:10.1073/pnas.0601852103

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., & Gerber, G. K. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, *298*(5594), 799–804. doi:10.1126/science.1075090

Li, H., Chen, H., Bao, L., Manly, K. F., Chesler, E. J., & Lu, L. (2006a). Integrative genetic analysis of transcription modules: towards filling the gap between genetic loci and inherited traits. *Human Molecular Genetics*, *15*(3), 481–492. doi:10.1093/hmg/ddi462

Li, H., Lu, L., Manly, K. F., Chesler, E. J., Bao, L., & Wang, J. (2005). Inferring gene transcriptional modulatory relations: A genetical genomics approach. *Human Molecular Genetics*, *14*(9), 1119–1125. doi:10.1093/hmg/ddi124

Li, Y., Alvarez, O. A., Gutteling, E. W., Tijsterman, M., Fu, J., & Riksen, J. A. (2006b). Mapping determinants of gene expression plasticity by genetical genomics in C. elegans. *PLOS Genetics*, *2*(12), e222. doi:10.1371/journal.pgen.0020222

Liu, B., de la Fuente, A., & Hoeschele, I. (2008). Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics*, *178*, 1763–1776. doi:10.1534/genetics.107.080069

Lucas, J., Carvalho, C., Wang, Q., Bild, A., & West, M. (2006). Sparse statistical modeling in gene expression genomics. In P. M. a. M. V. e. KA Do (Eds.), *Bayesian inference for gene expression and proteomics*. Cambridge: Cambridge University Press.

Lum, P. Y., Chen, Y., Zhu, J., Lamb, J., Melmed, S., & Wang, S. (2006). Elucidating the murine brain transcriptional network in a segregating mouse population to identify core functional modules for obesity and diabetes. *Journal of Neurochemistry*, *97*(Suppl 1), 50–62. doi:10.1111/j.1471-4159.2006.03661.x

Luo, Z. W., Potokina, E., Druka, A., Wise, R., Waugh, R., & Kearsey, M. J. (2007). SFP genotyping from affymetrix arrays is robust but largely detects cis-acting expression regulators. *Genetics*, *176*(2), 789–800. doi:10.1534/genetics.106.067843

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A., & Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, *431*(7006), 308–312. doi:10.1038/nature02782

Madigan, D., & Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, *89*(428), 1535–1546. doi:10.2307/2291017

Mahler, M., Most, C., Schmidtke, S., Sundberg, J. P., Li, R., & Hedrich, H. J. (2002). Genetics of colitis susceptibility in IL-10-deficient mice: Backcross versus F2 results contrasted by principal component analysis. *Genomics*, *80*(3), 274–282. doi:10.1006/geno.2002.6840

Mancosu, G., Pieroni, E., Maggio, F., Fotia, G., Liu, B., Hoeschele, I., et al. (2008). Deciphering a genome-wide yeast gene network. *Submitted*.

Mangin, B., Thoquet, P., & Grimsley, N. H. (1998). Pleiotropic QTL analysis. *Biometrics*, *54*, 88–99. doi:10.2307/2533998

McCulloch, G. E. R. (1996). Stochastic search variable selection. In S. R. a. D. S. WR Gilks (Ed.), *Markov chain Monte Carlo in practice* (pp. pp. 203-214). Boca Raton, FL: Chapman & Hall.

Medugorac, I., & Soller, M. (2001). Selective genotyping with a main trait and a correlated trait. *Journal of Animal Breeding and Genetics*, *118*(5), 285–295. doi:10.1046/j.1439-0388.2001.00308.x

Mnaimneh, S., Davierwala, A. P., Haynes, J., Moffat, J., Peng, W. T., & Zhang, W. (2004). Exploration of essential gene functions via titratable promoter alleles. *Cell*, *118*(1), 31–44. doi:10.1016/j.cell.2004.06.013

Murphy, K., & Mian, S. (1999). *Modelling gene expression data using dynamic Bayesian networks* (Tech. Rep.). Computer Science Division, University of California, Berkeley, CA.

Nadeau, J. H., Burrage, L. C., Restivo, J., Pao, Y. H., Churchill, G., & Hoit, B. D. (2003). Pleiotropy, homeostasis, and functional networks based on assays of cardiovascular traits in genetically randomized populations. *Genome Research*, *13*(9), 2082–2091. doi:10.1101/gr.1186603

Neale, M. C., Boker, S. M., Xie, G., & Maes, H. H. (2003). *Mx: Statistical modeling*. Richmond, VA: Department of Psychiatry.

Newman, M. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256. doi:10.1137/S003614450342480

Omholt, S. W., Plahte, E., Oyehaug, L., & Xiang, K. (2000). Gene regulatory networks generating the phenomena of additivity, dominance and epistasis. *Genetics*, *155*(2), 969–980.

Parkhomenko, E., Tritchler, D., & Beyene, J. (2007). Genome-wide sparse canonical correlation of gene expression with genotypes. *BMC Proceedings*, *1*(Suppl 1), S119. doi:10.1186/1753-6561-1-s1-s119

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge University Press.

Peck, S. C. (2005). Update on proteomics in Arabidopsis. Where do we go from here? *Plant Physiology*, *138*(2), 591–599. doi:10.1104/pp.105.060285

Pérez-Enciso, M., Quevedo, J. R., & Bahamonde, A. (2007). Genetical genomics: Use all data. *BMC Genomics*, *8*(69).

Pieroni, E., de la Fuente van Bentem, S., Mancosu, G., Capobianco, E., Hirt, H., & de la Fuente, A. (2008). Protein networking: insights into global functional organization of proteomes. *Proteomics*, *8*(4), 799–816. doi:10.1002/pmic.200700767

Raftery, A. E. (1993). Bayesian model selection in structural equation models. In K. A. Bollen & J. S. Long (Eds.), *Testing structural equation models* (pp. 163-180). Beverly Hills: Sage.

Richardson, T. (1996). *A polynomial-time algorithm for deciding Markov equivalence of directed cyclic graphical models.* Paper presented at the Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence, Portland, Oregon.

Richardson, T., & Spirtes, P. (1999). Automated discovery of linear feedback models. In C. Glymour & G. F. Cooper (Eds.), *Computation, causation, and discovery* (pp. 253-304). Cambridge, MA: MIT Press.

Rockman, M. V. (2008). Reverse engineering the genotype-phenotype map with natural genetic variation. *Nature*, *456*(7223), 738–744. doi:10.1038/nature07633

Rockman, M. V., & Kruglyak, L. (2006). Genetics of global gene expression. *Nature Reviews. Genetics*, *7*(11), 862–872. doi:10.1038/nrg1964

Ronald, J., Brem, R. B., Whittle, J., & Kruglyak, L. (2005). Local regulatory variation in *Saccharomyces cerevisiae*. *PLOS Genetics*, *1*(2), e25. doi:10.1371/journal.pgen.0010025

Rosa, G. J., de Leon, N., & Rosa, A. J. (2006). Review of microarray experimental design strategies for genetical genomics studies. *Physiological Genomics*, *28*(1), 15–23. doi:10.1152/physiolgenomics.00106.2006

Rostoks, N., Borevitz, J. O., Hedley, P. E., Russell, J., Mudie, S., & Morris, J. (2005a). Single-feature polymorphism discovery in the barley transcriptome. *Genome Biology*, *6*(6), R54. doi:10.1186/gb-2005-6-6-r54

Rostoks, N., Mudie, S., Cardle, L., Russell, J., Ramsay, L., & Booth, A. (2005b). Genome-wide SNP discovery and linkage analysis in barley based on genes responsive to abiotic stress. *Molecular Genetics and Genomics*, *274*(5), 515–527. doi:10.1007/s00438-005-0046-z

Roy, S. N. (1957). *Some aspects of multivariate analysis*. New York: Wiley.

Schadt, E., Lamb, J., Yang, X., Zhu, J., Edwards, S., & Guhathakurta, D. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, *37*(7), 710. doi:10.1038/ng1589

Schadt, E. E., Monks, S. A., Drake, T. A., Lusis, A. J., Che, N., & Colinayo, V. (2003). Genetics of gene expression surveyed in maize, mouse and man. *Nature*, *422*(6929), 297–302. doi:10.1038/nature01434

Schäfer, J., & Strimmer, K. (2005a). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics (Oxford, England)*, *21*, 754–764. doi:10.1093/bioinformatics/bti062

Schäfer, J., & Strimmer, K. (2005b). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, *4*, 32. doi:10.2202/1544-6115.1175

Schwikowski, B., Uetz, P., & Fields, S. (2000). A network of protein-protein interactions in yeast. *Nature Biotechnology*, *18*(12), 1257–1261. doi:10.1038/82360

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., & Koller, D. (2003). Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, *34*(2), 166–176. doi:10.1038/ng1165

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, *31*(1), 64–68. doi:10.1038/ng881

Shipley, B. (2002). *Cause and correlation in biology: A user's guide to path analysis, structural equations, and causal inference*. Cambridge University Press.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causation, prediction, and search*. MIT Press.

Spirtes, P., Glymour, C., Scheines, R., Kauffman, S., Aimale, V., & Wimberly, F. (2000). *Constructing Bayesian network models of gene expression networks from microarray data.* Paper presented at the Proc. Atlantic Symp. Comp. Biol., Genome Inf. Syst., and Technol.

Stein, C. M., Song, Y., Elston, R. C., Jun, G., Tiwari, H. K., & Iyengar, S. K. (2003). Structural equation model-based genome scan for the metabolic syndrome. *BMC Genetics*, *4*(Suppl 1), S99. doi:10.1186/1471-2156-4-S1-S99

Storey, J., Akey, J., & Kruglyak, L. (2005). Multiple locus linkage analysis of genomewide expression in yeast. *PLoS Biology*, *3*(8), e267. doi:10.1371/journal.pbio.0030267

Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(16), 9440–9445. doi:10.1073/pnas.1530509100

Stylianou, I. M., Korstanje, R., Li, R., Sheehan, S., Paigen, B., & Churchill, G. A. (2006). Quantitative trait locus analysis for obesity reveals multiple networks of interacting loci. *Mammalian Genome*, *17*(1), 22–36. doi:10.1007/s00335-005-0091-2

Thaller, G., & Hoeschele, I. (2000). Fine-mapping of quantitative trait loci in half-sib families using current recombinations. *Genetical Research*, *76*, 87–104. doi:10.1017/S0016672300004638

Threadgill, D. W. (2006). Meeting report for the 4th Annual Complex Trait Consortium meeting: From QTLs to systems genetics. *Mammalian Genome*, *17*(1), 2–4. doi:10.1007/s00335-005-0153-5

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Methodological*, *58*(1), 267–288.

Tong, A. H., Lesage, G., Bader, G. D., Ding, H., Xu, H., & Xin, X. (2004). Global mapping of the yeast genetic interaction network. *Science*, *303*(5659), 808–813. doi:10.1126/science.1091317

Tu, Z., Wang, L., Arbeitman, M. N., Chen, T., & Sun, F. (2006). An integrative approach for causal gene identification and gene regulatory pathway inference. *Bioinformatics (Oxford, England)*, *22*(14), e489–e496. doi:10.1093/bioinformatics/btl234

Vuylsteke, M., van Eeuwijk, F., Van Hummelen, P., Kuiper, M., & Zabeau, M. (2005). Genetic analysis of variation in gene expression in Arabidopsis thaliana. *Genetics*, *171*(3), 1267–1275. doi:10.1534/genetics.105.041509

Waaijenborg, S., Verselewel de Witt Hamer, P. C., & Zwinderman, A. H. (2008). Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis. *Stat Appl Genet Mol Biol*, *7*, Article3.

Wagner, A. (2001). How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. *Bioinformatics (Oxford, England)*, *17*(12), 1183–1197. doi:10.1093/bioinformatics/17.12.1183

Wagner, A., & Fell, D. A. (2001). The small world inside large metabolic networks. *Proceedings. Biological Sciences*, *268*(1478), 1803–1810. doi:10.1098/rspb.2001.1711

Wang, C. (2007). Variational Bayesian approach to canonical correlation analysis. *IEEE Transactions on Neural Networks*, *18*, 905–910. doi:10.1109/TNN.2007.891186

Wang, D., & Nettleton, D. (2006). Identifying genes associated with a quantitative trait or quantitative trait locus via selective transcriptional profiling. *Biometrics, 62*(2), 504–514. doi:10.1111/j.1541-0420.2005.00491.x

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. doi:10.1038/30918

Werhli, A. V., & Husmeier, D. (2007). Reconstructing gene regulatory networks with bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology, 6*, 15. doi:10.2202/1544-6115.1282

West, M. A., Kim, K., Kliebenstein, D. J., van Leeuwen, H., Michelmore, R. W., & Doerge, R. W. (2007). Global eQTL mapping reveals the complex genetic architecture of transcript-level variation in Arabidopsis. *Genetics*, *175*(3), 1441–1450. doi:10.1534/genetics.106.064972

Wille, A., & Buhlmann, P. (2006). Low-order conditional independence graphs for inferring genetic networks. *Stat. Appl. Genet. Mol. Biol., 5*(1), Article 1.

Wille, A., Zimmermann, P., Vranova, E., Furholz, A., Laule, O., & Bleuler, S. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome Biology*, *5*(11), R92. doi:10.1186/gb-2004-5-11-r92

Xiong, M., Li, J., & Fang, X. (2004). Identification of genetic networks. *Genetics*, *166*(2), 1037–1052. doi:10.1534/genetics.166.2.1037

Yi, N., George, V., & Allison, D. B. (2003). Stochastic search variable selection for identifying multiple quantitative trait loci. *Genetics, 164*(3), 1129–1138.

Yvert, G., Brem, R. B., Whittle, J., Akey, J. M., Foss, E., & Smith, E. N. (2003). Transacting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors. *Nature Genetics*, *35*(1), 57–64. doi:10.1038/ng1222

Zeng, Z. (1993). Theoretical basis for separation of multiple linked gene effects in mapping quantitative trait loci. *Proceedings of the National Academy of Sciences of the United States of America*, *90*(23), 10972–10976. doi:10.1073/pnas.90.23.10972

Zeng, Z. B. (1994). Precision mapping of quantitative trait loci. *Genetics*, *136*(4), 1457–1468.

Zeng, Z.-B., Liu, J., Stam, L. F., Kao, C.-H., Mercer, J. M., & Laurie, C. C. (2000). Genetic architecture of a morphological shape difference between two *Drosophila* species. *Genetics*, *154*(1), 299–310.

Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology, 4*, 17. doi:10.2202/1544-6115.1128

Zhou, L., Mideros, S. X., Bao, L., Tripathy, S., Torto-Alalibo, T. A., Mao, Y., et al. (2008). *Dissecting soybean resistance to Phytophthora by QTL analysis of host and pathogen expression profiles.* Paper presented at the International Plant and Animal Genome Conference XVI, San Diego.

Zhu, J., Lum, P. Y., Lamb, J., Guha Thakurta, D., Edwards, S. W., & Thieringer, R. (2004). An integrative genomics approach to the reconstruction of gene networks in segregating populations. *Cytogenetic and Genome Research*, *105*(2-4), 363–374. doi:10.1159/000078209

Zhu, J., Wiener, M. C., Zhang, C., Fridman, A., Minch, E., & Lum, P. Y. (2007). Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. *PLoS Computational Biology*, *3*(4), e69. doi:10.1371/journal.pcbi.0030069

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B. Methodological*, *67*, 301–320. doi:10.1111/j.1467-9868.2005.00503.x

KEY TERMS AND DEFINITIONS

Bayesian Network: Bayesian networks are directed probabilistic graphical models that represent conditional independence relationships among variables of interest.

eQTL: In Genetical Genomics, the gene expression levels are considered as phenotypic traits. Therefore, the identified QTLs are referred to as 'expression-QTLs' or 'eQTLs'.

etrait: In Genetical Genomics, the gene expression levels are considered as phenotypic traits. Therefore, we call gene expression levels as 'expression traits' or in short 'etraits'.

False Discovery Rate: False Discovery Rate (FDR) is the expected false positive rate in multiple hypothesis testing. Among the list of rejected hypotheses, FDR controls the expected proportion of incorrectly rejected null hypotheses.

Family-Wise-Error Rate: Family-Wise-Error Rate (FWER) (also referred to as the genome-wise error rate in the context of QTL mapping) is the probability of making one or more false discoveries in multiple hypothesis testing. FWER control is more conservative (and less powerful) than FDR control.

Genetical Genomics: Genetical Genomics, also referred to as 'the genetics of gene expression', uses naturally occurring, multi-factorial perturbations in segregating or genetically randomized populations. Genetical Genomics approaches integratively analyze gene expression data and genotype data (measurable DNA sequence polymorphisms) and make use of DNA sequence information when available.

Quantitative Trait Locus: Quantitative trait locus (QTL) is a chromosomal region that causally affects a phenotypic trait under consideration. Statistically, a QTL is a confidence interval for the genomic location of a DNA polymorphism that is causal for the phenotype of interest.

Structural Equation Modeling: Structural Equation Modeling is a linear statistical modeling framework for testing and estimating causal relationships among variables. It has been widely used in econometrics, sociology and psychology, usually as a confirmatory procedure instead of an exploratory analysis for causal inference.

Chapter 5
Inferring Genetic Regulatory Interactions with Bayesian Logic-Based Model

Svetlana Bulashevska
German Cancer Research Centre (DKFZ), Germany

ABSTRACT

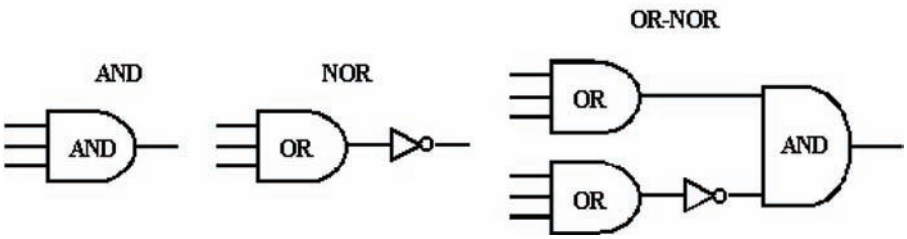
This chapter describes the model of genetic regulatory interactions. The model has a Boolean logic semantics representing the cooperative influence of regulators (activators and inhibitors) on the expression of a gene. The model is a probabilistic one, hence allowing for the statistical learning to infer the genetic interactions from microarray gene expression data. Bayesian approach to model inference is employed enabling flexible definitions of a priori probability distributions of the model parameters. Markov Chain Monte Carlo (MCMC) simulation technique Gibbs sampling is used to facilitate Bayesian inference. The problem of identifying actual regulators of a gene from a high number of potential regulators is considered as a Bayesian variable selection task. Strategies for the definition of parameters reducing the parameter space and efficient MCMC sampling methods are the matter of the current research.

INTRODUCTION

The advent of microarray technology facilitated monitoring of gene expression and posed the problem of reconstructing genetic regulatory relations from data. A concept of *gene regulatory network* evolved, as a graphical representation of interactions between genes. This is a simplification of the underlying molecular biological regulatory mechanism, since the expression levels of some genes affect the expression of other genes indirectly, via the synthesis of proteins, protein complex formation, DNA binding etc. Mathematical models of genetic regulatory networks define features of the regulation by means of mathematical functions and propose algorithms in order to infer network models (i.e. connectivity, parameters etc.) from experimental data.

DOI: 10.4018/978-1-60566-685-3.ch005

Figure 1. Examples of genetic regulatory functions presented as logic gates



The attempt to model genetic regulation was pioneered long before the appearance of high-throughput molecular genetics methods (Kauffman 1969, 1996). It was stated that the regulatory interactions between genes can be presented as logic gates as exemplified in Figure 1, and the Boolean network model was proposed. In the Boolean network, discrete states of genes (the active and the not active) are admitted, and the state of each gene is functionally determined by the states of some other genes using the rules of logics. Continuous gene expression measurements must be discretized before they can be used for Boolean network modeling.

The fundamental idea behind the Boolean network is that the gene regulation is executed by transcription factors transcribed from a number of genes, which cooperatively bind to the binding sites of a target gene. This constitutes a so called *cis-regulatory element*, the working principles of which can be described by means of logics. Some genes are activated by one of several different possible transcription factors ("OR" logic). Other genes require that two or more transcription factors must all be bound for the activation ("AND" logic). The activation of some genes may be inhibited by one of a few possible repressor proteins ("NOT OR" logic, in our notation "NOR"). Further on, in case of "OR-NOR" logic, a gene is regulated by a set of possible activators and a set of possible inhibitors. The gene is transcribed if and only if one of its possible activators is active and it is not repressed by one of its possible repressors. An algorithm REVEAL was developed to reverse-engineer Boolean logic relations from expression data, based on mutual information between input and output states (Somogyi and Sniegosky, 1996; Liang et al., 1998). The major limitation of the Boolean network model was its inherent determinism, which contradicts with the stochastic nature of the underlying process of gene regulation and limits the reliability of relations inferred from real data.

Later on, extensions of Boolean Networks were suggested to make them robust against noise. In the *noisy Boolean networks* of Akutsu (2000), a certain probability is defined, with which a number of input/output patterns will not be discarded by an inference algorithm, even if a Boolean function is not satisfied. In the *Probabilistic Boolean Networks* (Shmulevich et al. 2002), more than one Boolean function are defined for each gene, and the particular function for calculating the state of the gene is selected with a certain probability.

Friedman et al. (2000) were the first to employ *probabilistic graphical models*, particularly *Bayesian networks*, to model genetic regulatory network. Probabilistic (statistical) modeling uses probability distributions to describe the states of the modeling variables and their dependencies. Probabilistic graphical models (Jordan, 2004) are graphs in which nodes represent random variables, and the missing edges between the nodes represent conditional independencies among the variables. In this way, the *joint probability distribution* of the variables is represented in a compact form. This reduces the number of parameters needed to describe the whole probabilistic model and sets a basis for statistical inference. Bayesian

network (Pearl, 1998; Jensen, 1996) is a common type of the probabilistic graphical models, where the graph is directed and acyclic (DAG). The graph encodes conditional independencies as follows: given the value of its parents in the graph, the variable is conditionally independent of other variables except its descendants. Then, the joint probability distribution of the variables factorizes into the product of the *conditional probability distributions* (CPD). The CPD for a variable defines its probability given every possible combination of the values of its parents. Thus, the state of a gene is described as a probability distribution dependent on a set of its immediate regulators. The global relations of genes in the genetic regulatory network can be described as being composed of local interactions between each gene and its regulatory genes. The learning of a Bayesian network from data comprises two tasks: the graphical structure learning and the estimation of the parameters of the conditional distributions.

The drawback of the Bayesian network approach is that the combinatorial semantics of the interaction of parents makes it difficult to interpret the results of network learning and to uncover the "true" *cis*-regulatory relationships covered in this presentation.

In this chapter, a model for genetic regulatory interactions is presented that combines the simple and biologically motivated Boolean logic semantics of Boolean networks and the possibility of dealing with uncertainty offered, in particular, by Bayesian networks, and, in general, by the Bayesian statistical modelling. The model is a special case of the Bayesian network in that the local probability distributions are constrained to noisy logic functions. The model can be seen as an intermediate between the local models of interactions, defined in Boolean networks, and Bayesian networks.

The chapter describes a statistical learning approach that allows for a particular gene to find a set of its regulators (activators and inhibitors), given a particular Boolean logic function governing this regulation. To robustly identify the regulators of a target gene from a large number of potential regulators is a great challenge in view of the sparseness of experimental data. The Bayesian learning framework and appropriate formulations of a priori distributions of network parameters presented here allow for an efficient search over the space of possible models and penalization of complex models.

In the following, we give a brief introduction to the Boolean and the Bayesian Networks, and explain the Bayesian logic-based model. After the introduction to the Bayesian modelling and MCMC sampling-based approaches, the Bayesian learning of the model from data is described. The main idea is the Bayesian variable selection approach. Hints for the specification of the parameters hereto are recommended. We demonstrate the application of the model exemplary on the malaria parasite data. Further related approaches are discussed, completed with the conclusion and the outlook for future research.

MAIN THRUST OF THE CHAPTER

Boolean Network

A Boolean network is a system of n binary-state nodes. Each node is assigned regulatory inputs from several other nodes and a Boolean function, according to which the state of the node is computed from the input states. Each Boolean function is specified with a *truth table*. For instance, Table 1 displays the truth table for the Boolean function "OR".

The state of a network at a time point t is given by the current states of all the n nodes. Thus the state space of any such network is 2^n. *Simulation* is executed in discrete time steps $\ldots, t, t + 1, \ldots$, where each node obtains its new state according to the inputs. Since the Boolean Network has a limited number of

Table 1. Truth table for the Boolean function "OR" used in the Boolean Network

X_1	X_2	Y
0	0	0
1	0	1
0	1	1
1	1	1

possible states, it will reach a previously visited state, and hence, due to the deterministic dynamics, will fall into an *attractor*. For a given set of inputs, the attractor reached is called *logical steady state*. It gives an impression about what network state is possible under the fixed states.

Bayesian Network

The Bayesian network comprises two components: the qualitative one and the quantitative. The qualitative component is a directed acyclic graph G, whose vertices correspond to the random variables $X_1,\ldots,X_n$. The graph G encodes *conditional independencies* between the variables: given the value of its parents in G, the variable is conditionally independent of other variables in the network except its descendants. Due to this, the joint probability distribution is equal to the product of the *conditional probability distributions* (CPDs):

$$P(X_1,\ldots,X_n) = \prod_{i=1,\ldots,n} P(X_i \left| parents(X_i)\right).$$

In other words, the joint distribution can be represented in a factorized form. The conditional distribution for a variable defines its probability given every possible combination of its parents' values. Due to the notion of conditional independence, probabilistic dependencies among the variables in the network can be represented only by the specification of CPDs. The set of all CPDs is the quantitative component of the Bayesian network. In fact, the CPD is the multinomial distribution with parameters $\boldsymbol{\alpha}$ (vector). The CPDs are specified with the so called *conditional probability tables* (CPTs). Figure 2 presents an example of a Bayesian Network with seven variables.

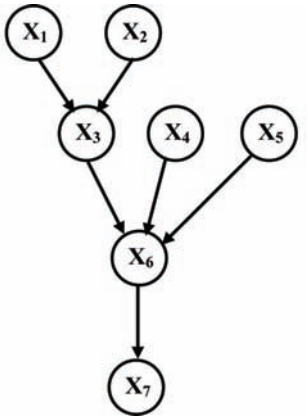
The joint probability distribution of the variables factorizes into:

$$P(X_1,\ldots,X_7) = P(X_1)P(X_2)P(X_3 \left| X_1, X_2\right) P(X_4)P(X_5)P(X_6 \left| X_3, X_4, X_5\right) P(X_7 \left| X_6\right).$$

The specification of a CPT, with four parameters $\alpha_1,\ldots,\alpha_4$ for the variable X_3 having two parents X_1 and X_2, is displayed in Table 2.

When inferring a Bayesian network from observational data, each candidate network must be scored, based on its ability to explain the data. Therefore, scoring metrics are used. Two different approaches exist to derive the scoring metrics: a maximum likelihood-based and a fully Bayesian. In the former, the best fit to data D for a given DAG G is determined by maximizing the likelihood $p(D|G, \boldsymbol{\alpha})$ as a function of $\boldsymbol{\alpha}$, the parameters of the conditional probability distributions. A score is then given by:

Figure 2. Example of a Bayesian network



$$score_{ML}(G) = \max_{\pm} p(D \mid G, \pm).$$

Since this score tends to over-fitting, the BIC score (*Bayesian information criterion*) is often used, penalizing the maximum likelihood of the model with respect to the number of parameters (Schwarz, 1978).

In the Bayesian approach, the posterior probability of model structure *G* given data *D* is evaluated:

$$score_{Bayes}(G) = p(G \mid D) = \frac{p(D \mid G)p(G)}{p(D)}.$$

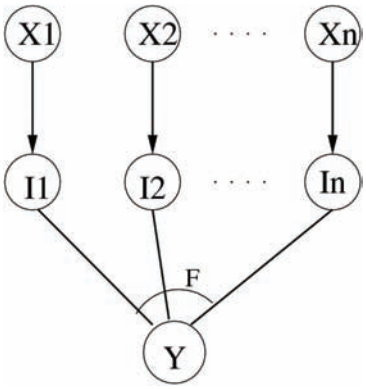
Here, $p(D|G)$ is the *marginal likelihood* and $p(G)$ is a *prior* over model structure. The denominator $p(D)$ is called a *normalizing constant* and is the same for all models, so one does not need to compute it for the scoring. In the marginal likelihood, the parameters $\boldsymbol{\alpha}$ are being integrated out (and not maximized), that precludes over-fitting:

$$p(D \mid G) = \int_{\pm} p(D \mid G, \pm)p(\pm \mid G)d\pm.$$

Table 2. Conditional probability table specifying the conditional distribution of the variable X_3 given its parents X_1 and X_2

X_3			
X_1	X_2	0	1
0	0	α_1	$1-\alpha_1$
1	0	α_2	$1-\alpha_2$
0	1	α_3	$1-\alpha_3$
1	1	α_4	$1-\alpha_4$

Figure 3. Bayesian logic-based model of gene regulatory interactions, F being a Boolean function ("AND", "OR")



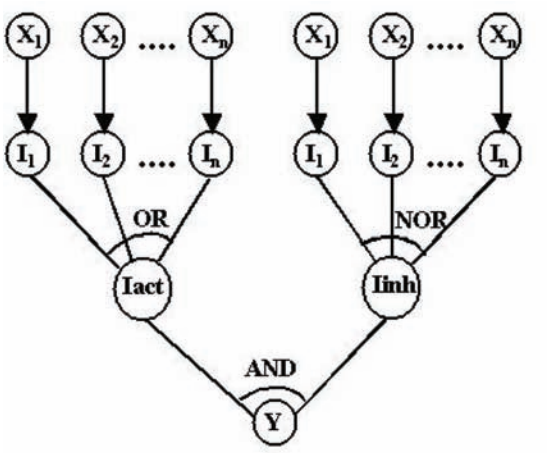
This integral can be computed analytically, if the prior probability distribution of the parameters is chosen in a special way, namely *conjugate* to the likelihood (see also explanations below). Using that the Dirichlet distribution is the conjugate prior for the multinomial, the *Bayesian score* for the Bayesian network was derived (Heckerman, 1998). The prior for the structure $p(G)$ can help to penalize or to give preference to models with particular features, but the simplest choice is the uniform distribution assuming that the models are equally likely. Heuristic algorithms (*hill climbing*, *arc inversion*) are used to obtain the optimal Bayesian network structure with respect to the Bayesian score (Cooper and Herskovits, 1992).

The Bayesian network formalism allows to model arbitrary interactions between parents $X_1,\ldots,X_n$ of a variable Y. The complete CPD for a binary variable with n parents requires the specification of 2^n independent parameters (one parameter for each parents' state configuration). This combinatorial semantics of parents interaction in the Bayesian network, and hence the exponential explosion of the parameter space make the model learning computationally costly. Moreover, in small data sets there might be an insufficient number of cases available for learning conditional probabilities. It is more reliable to learn distributions having fewer parameters. These considerations motivated the employment of a further probabilistic graphical model with a constrained definition of the local probability distribution, in order to adequately model the genetic regulatory control. The model will be presented in the next section.

The Bayesian Logic-Based Model of Gene Regulatory Interactions

As previously discussed, in graphical modelling, the joint probability distribution of variables is expressed as a product of distributions over a smaller number of variables by exploiting conditional independence relations encoded in a graph structure. In this way, the number of parameters to be specified or estimated is reduced. For example, in the Bayesian network formalism compact representation of the joint distribution among variables in the network is achieved by expressing it with conditional probability distributions. One can further exploit the independencies between parents of a variable in a Bayesian network to get more compact representations of CPDs. In the past, several models were proposed with special types of causal interaction (see Heckerman and Breese, 1994; Meek and Heckerman, 1997; Srinivas, 1993). One is the *causal independence* model which assumes independence of parents of each variable

Figure 4. Complex model of gene regulatory interactions with activators and inhibitors ("OR-NOR" regulation)



in the model. The variables $X_1,\ldots,X_n$, the parents of the variable Y, can influence Y through independent "mechanisms". The effects are then combined by a rule determined by a Boolean-logic function. Such models were introduced originally by J.Pearl (1998) and called "noisy OR-Gate" and "noisy AND-Gate". This kind of models is employed here for modelling the genetic regulatory interactions.

We assume that a variable X_i (*regulator*) can execute its influence on variable Y (*regulatee*) independently of other possible regulators from the set $X_1,\ldots,X_n$. The biological mechanism underlying this modelling assumption is the binding of protein transcribed by the regulator to the DNA of the regulatee. This process is not deterministic, rather, each gene X_i can regulate Y with probability θ_i and can fail to do this with probability $1\text{-}\theta_i$. The model is represented by a directed graph in Figure 3. In the model, intermediate variables $I_1,\ldots,I_n$ were introduced, through which the variables $X_1,\ldots,X_n$ exert their influence on a given common effect variable Y.

Each intermediate variable I_i has only one parent, the variable X_i. Its probability distribution is defined as follows: given that X_i=1, I_i takes the value 1 with probability θ_i and the value 0 with probability $1\text{-}\theta_i$, respectively. Given that X_i=0, I_i takes the value 0 with probability 1. The combined regulatory influence on the variable *Y* is calculated as the Boolean function *F* on the input variables $I_1,\ldots,I_n$. If $X_1,\ldots,X_n$ are activators, then the state of the variable *Y* is $F(I_1,\ldots,I_n)$; if $X_1,\ldots,X_n$ are inhibitors, the state of *Y* is

Table 3. Conditional probability table of regulatee Y that is activated by two regulators X_1 and X_2 ("OR"-activation)

Y			
X_1	X_2	0	1
0	0	1	0
1	0	$1\text{-}\theta_i$	θ_i
0	1	$1\text{-}\theta_2$	θ_2
1	1	$(1\text{-}\theta_1)(1\text{-}\theta_2)$	$1\text{-}(1\text{-}\theta_1)(1\text{-}\theta_2)$

1-$F(I_1,\ldots,I_n)$. The Boolean "interaction function" F defines in which way the intermediate effects I_i, and indirectly the variables X_i, interact. We consider two interaction functions: AND and OR. The semantics of the OR-function implies that the variables X_i are each assumed to be sufficient to influence Y. In the case of the AND-function, all variables X_i need to execute their own influence on the variable Y in order Y to be active.

The introduction of the hidden state variables I_i allows for the insertion of "noise" into the Boolean-logic based models. It allows to model that the biological mechanism of the regulation of one gene by another could be inhibited for unknown reasons. Thus, the input variables can be considered as observables from which we make our measurements, while the hidden variables have the "true" latent biological values.

In the present chapter, we consider simple models with activatory regulation ("OR", "AND") and inhibitory regulation ("NOR", "NAND"), as well as complex models: "AND-NAND", "AND-NOR", "OR-NAND" and "OR-NOR". In the complex models, the regulatory influences of multiple activators and multiple inhibitors are combined with AND-function as displayed in Figure 4.

The conditional probability distribution for the regulatee Y that is activated by two possible regulators ("OR"-activation), is presented in Table 3. Note that the model with the Boolean logic-based interaction of parent variables allows for the specification of the CPD for a variable with only n parameters $\theta_1,\ldots,\theta_n$, i.e. polynomial on the number of parents.

We formulate the problem of learning the model from data as follows: given the data on gene Y and its potential regulators $X_1,\ldots,X_n$, for a given Boolean logic function F, identify the subset $X_1,\ldots,X_n$ of actual regulators of Y. The parameters θ must also be assessed. We employ statistical learning of the model from data, in particular, the Bayesian inference.

Bayesian Inference

In contrast to the classical frequentist approach, Bayesian inference does not deal with point estimates of model parameters, but, rather, with probability distributions on the parameters and on all unobserved quantities (such as latent variables, predictions etc.). This enables to assess a whole interval as having a high probability of containing an unknown quantity of interest.

Bayesian modelling starts with setting up a *full probability model* – a joint probability distribution for all observed and unobserved quantities in a problem. Then, the Bayesian methodology seeks to assess the conditional probability distributions of the unobserved quantities given the observed data. Let θ stands for unobservable quantities (parameters) and y for observable (data). Then, the joint probability is $p(\theta,y)$ and the posterior probability by Bayes' rule is:

$$p(\theta \mid y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y \mid \theta)p(\theta)}{p(y)},$$

where $p(y|\theta)$ is called *likelihood function* and $p(\theta)$ is the *a priori* probability of the parameters. Since $p(y)$ does not depend on θ, the following proportionality is valid: $p(\theta \mid y) \propto p(y \mid \theta)p(\theta)$. This '$posterior \propto likelihood \times prior$' rule is the basis of the Bayesian inference. The $p(y)$ is the *marginal likelihood* integrated over the parameters θ: $p(y) = \int_\theta p(y \mid \theta)p(\theta)d\theta$.

The calculation of the multi-dimensional integrals arising in the Bayesian inference is the general computational obstacle of the Bayesian methodology. The integrals are analytically tractable only in certain restricted examples. When the posterior distribution of the parameters belongs to the same parametric family as the prior distribution, the integral has a closed form solution. This property is called *conjugacy* (Bernardo and Smith, 1994). For example, conjugate priors are available for the *general exponential family* models. Alternatively, approximation techniques, such as *variational* methods (Jaakkola, 1997) or simulation *Markov Chain Monte Carlo* (MCMC) (Gilks 1993) can be employed. For an introduction into the MCMC see Gamerman (2006).

One of the MCMC approaches is known as *Gibbs sampling* (Geman and Geman, 1984). It reduces the problem of dealing simultaneously with a large number of unknown parameters in a joint distribution into a much simpler problem of dealing with one variable at a time, iteratively sampling each from its full conditional distribution given the current values of all other variables in the model. It happens that many models have a complex joint distribution, but their conditional distributions are relatively simple. As stated by Pearl (1987), performing Gibbs sampling is particularly appropriate for a graphical model due to the factorization of the joint probability.

A specification of the prior distribution for parameter θ should make all its possible values equally probable (*non-informative* prior). This guarantees that the prior distribution plays a minimal role in the posterior, and the whole parameter space will be explored. However, a "subjective" definition of the prior is possible, when a desire is to insert a priori knowledge into the model. This possibility is an inherent advantage of the Bayesian modelling over classical statistical approaches.

Further more, Bayesian modelling allows for a hierarchical formulation of a model: distributions for the parameters can be formulated, in turn, with the help of *hyperparameters*. This provides a great flexibility in defining complex models fitting them more adequately to real domains.

Bayesian Model Selection

Our task is to infer from data not only parameters of the model, but the structure of the model itself. In the Bayesian framework, this task is often called *Bayesian model selection*. As we have seen previously on the example of the Bayesian network, the problem is addressed by calculating the posterior probability of a model given data for a collection of candidate models and selecting the most probable model. Suppose that the data D has been generated by a model m , one of a set M of candidate models, $m \in M$. If $p(m)$ is the prior probability of model m, then the posterior model probability by Bayes rule is $p(m \mid D) \propto p(D \mid m)p(m)$. Let θ_m be parameters of the model m. The marginal likelihood $p(D|m)$ is calculated as:

$$p(D \mid m) = \int p(D \mid m, \theta_m) p(\theta_m \mid m) d\theta_m,$$

where $p(\theta_m|m)$ is the prior distribution of model parameters θ_m for model m. When the integral has no analytical solution, MCMC can be employed. MCMC samples from the joint posterior distribution $p(m,\theta_m|D)$ allowing for the estimation of the posterior model probability $p(m|D)$ and of the posterior parameter probability $p(\theta_m|D)$.

We now proceed with the Bayesian formulation of the Boolean-logic based models. Consider the model with "OR"-activation. Assume the variable *Y* is commonly influenced by the variables $X_1,\ldots,X_n$. The probability distribution of Y given the values of its parents can be written as:

$$P(Y = 0 \mid \theta) = \prod_{i=1}^{n} (1-\theta_i)^{X_i} \text{ and}$$

$$P(Y = 1 \mid \theta) = 1 - \prod_{i=1}^{n} (1-\theta_i)^{X_i},$$

where $\theta = (\theta_1,\ldots,\theta_n)$ is the vector of parameters. Assume we have a sample of *N* cases corresponding to the states of the variables $X_1,\ldots,X_n$ and *Y*. Denote by Y_j the state of the variable *Y* in case *j*, and by X_{ij} the state of the variable X_i in case *j*. The likelihood function is then:

$$L(\theta) = \prod_{j=1}^{N} (\prod_{i=1}^{n} (1-\theta_{ij})^{X_{ij}})^{1-Y_j} (1 - \prod_{i=1}^{n} (1-\theta_{ij})^{X_{ij}})^{Y_j}$$

If we substitute Ψ_{ij} by $-log(1-\theta_{ij})$, the likelihood function transforms into:

$$L(\psi) = \prod_{j=1}^{N} (e^{-\eta_j})^{1-Y_j} (1 - e^{-\eta_j})^{Y_j},$$

where $\eta_j = \sum_{i=1}^{n} \psi_{ij} X_{ij}$ is a linear predictor. This shows that the "OR"-model cannot be expressed in the exponential form. In fact, this is the *generalized linear model* (McCullagh and Nelder, 1983). Unlikely to exponential models, a straightforward conjugate prior for parameters (*regression coefficients*) is not available for this class of models. Chen and Ibrahim (2003) construct a prior based on a priori prediction on the response *Y*. Bedrick et al. (1996) developed *Data Augmentation Priors* based on evaluation of the prior at *n* locations in the predictor space. We instead turn to the MCMC.

We need to specify joint distribution for both data and parameters. The "OR" model can be written as:

$$Y \sim Bernoulli(1 - \prod_{i=1}^{n} (1-\theta_i)^{X_i})$$

(the operator ~ stands for 'is distributed as').

Now consider the complex model "OR-NOR". Assume the variable *Y* is influenced by a set of activators $X_1^{act},\ldots,X_n^{act}$ and a set of inhibitors $X_1^{inh},\ldots,X_k^{inh}$. The variable *Y* takes the value 1, if the activators executed their influence *and* the inhibitors failed, otherwise *Y* is 0. The "OR-NOR" model can then be defined as:

$$Y \sim Bernoulli((1 - \prod_{i=1}^{n} (1-\theta_{ij}^{act})^{X_{ij}^{act}}) \prod_{i=1}^{k} (1-\theta_{ij}^{inh})^{X_{ij}^{inh}})$$

In the following we show how the models can be reformulated to solve the problem of the model selection.

Bayesian Variable Selection

In our problem of Bayesian model selection, candidate models have different number of parameters (i.e. different numbers of regulatory genes). Because of the variable size of the problem space, standard Markov Chain Monte Carlo techniques cannot be directly applied. Essentially, two approaches exist. The first is a sophisticated simulation technique using Markov chain with jumps between the different models – *reversible jump MCMC* by Green (1995). Alternatively, all models under consideration are indexed and the index is treated as another parameter, to be considered jointly with all other model parameters. Carlin and Chib (1995) proposed this concept of a *supermodel* defined over a *composite parameter space* and used the standard MCMC methodology - Gibbs sampling. The algorithm was improved in the *Metropolized Carlin and Chib algorithm* (see Godsill (2001) and Dellaportas (2002)). Further Gibbs sampling approaches for model selection problems were developed by George and McCulloch (1996) – *Stochastic Search Variable Selection*, by Kuo and Mallick (1998), and by Dellaportas *et al.* (2000, 2002) – *Gibbs Variable Selection* (GVS).

The general idea is to substitute the model indicator $m \in M$ with a *variables indicator* $\gamma = (\gamma_1,\ldots,\gamma_n)$, a binary vector, representing which of the $X_j, j = 1,\ldots,p$ should be included in the desirable "true" model. This permits to consider one joint space of the model parameters and the variables indicator while keeping the dimensionality constant across all possible models. The model selection problem is then referred to as the *variable selection* problem.

Once the variables indicator has been introduced, the "OR" model is written as:

$$Y \sim Bernoulli(1 - \prod_{i=1}^{n} (1 - \theta_i)^{\gamma_i X_i})$$

The Bayesian approach requires setting up a joint probability distribution over all parameters $p(\theta,\gamma)$. Let D denote the observed data for the variables $X_j, j = 1,\ldots,p$ and Y. The joint posterior distribution given the observed data is $p(\theta,\gamma \mid D)$. The Gibbs sampling procedure samples successively from univariate conditional distributions, simulating a Markov chain

$$\theta^{(0)}, \gamma^{(0)}, \theta^{(1)}, \gamma^{(1)}, \ldots, \theta^{(t)}, \gamma^{(t)}, \ldots$$

which converges in distribution to $p(\theta,\gamma \mid D)$. The subsequence

$$\gamma^{(0)}, \gamma^{(1)}, \ldots, \gamma^{(t)}, \ldots$$

converges to $p(\gamma \mid D)$. This sequence can be used to identify the high probability values of γ_j which are the values that appear most frequently in the sequence. And this is namely the desirable result, indicating the true regulators of a target gene.

Consider a partition of θ into $(\theta_\gamma, \theta_{-\gamma})$ corresponding to those components of θ which are included and not included, respectively, in the model. Then the posterior distribution of the parameters $p(\theta,\gamma|D)$ may be partitioned into $p(\theta_\gamma|\theta_{-\gamma},\gamma,D)$ and $p(\theta_{-\gamma}|\theta_\gamma,\gamma,D)$. From the model definition it is obvious that the components of the vector $\theta_{-\gamma}$ do not affect the model likelihood. The full conditional posterior distributions required for the Gibbs sampling procedure are given by:

$$p(\theta_\gamma \mid \theta_{-\gamma}, \gamma, D) \propto p(D \mid \theta, \gamma) p(\theta_\gamma \mid \gamma) p(\theta_{-\gamma} \mid \theta_\gamma, \gamma),$$

$$p(\theta_{-\gamma} \mid \theta_\gamma, \gamma, D) \propto p(\theta_{-\gamma} \mid \theta_\gamma, \gamma),$$

where $p(D|\theta,\gamma)$ is the model likelihood, $p(\theta_\gamma|\gamma)$ is the model prior and $p(\theta_{-\gamma}|\theta_\gamma,\gamma,)$ is called *pseudoprior*.

Methods for Gibbs variable selection differ in their approaches on specifying prior distributions for the model parameters. The most simple is the "unconditional prior" approach of Kuo and Mallick where the prior distribution of model parameters θ is defined independent of variables indicator γ. In the Stochastic Search Variable Selection method of George and McCulloch, the priors for θ_j depend on γ_j and are defined as mixtures of two Normal distributions for $\gamma_j = 0$ and $\gamma_j = 1$. If $\gamma_j = 0$, the parameters (pseudopriors) are kept close to 0 by defining the mean of the normal distribution equal to 0. The method of Dellaportas *et al.* (2000, 2002) differs from the SSVS in that the pseudopriors may not be distributed around 0, instead they may be chosen in a way to help increase the efficiency of the sampling procedure. Carlin and Chib (1995) noted that the pseudoprior distributions are meaningless as a modelling device, but must be chosen carefully as they affect the rate of convergence of the chain. Total freedom may be given to the specification of pseudopriors, they may even include specifications using the data. It was recommended to set the pseudoprior distribution $p(\theta_j|\gamma_j=0)$ as close as possible to $p(\theta_j|\gamma_j=1)$ (*proposal* densities). Dellaportas *et al.* use these proposal densities which can be estimated using a *pilot run* of the MCMC for the *saturated* model, i.e. the model where all terms $\gamma_j=1$ for all j. The present approach adopts the method of Dellaportas *et al.* (2000, 2002).

Solutions and Recommendations

Discretization

When applying Boolean logic-based models, it is necessary to preprocess the continuous gene expression values and to *discretize* them into two states (0 - not expressed, 1 - expressed). Discretization results in a loss of information, however, it reduces noise, which is characteristic to the mRNA measurements, and makes the inference of the model more robust. To perform discretization, *vector quantization* techniques can be used such as the clustering algorithm *k-means* (Gersho and Gray, 1992). For example, for each gene, its expression values can be clustered into two groups ($k = 2$) with two initial values: 0 and the maximum expression value of the gene. Several statistically sound quantization approaches were proposed (Chung, 2006; Di Camillo, 2005). In contrast to the approaches which execute discretization before and independently of the model inference, Steck and Jaakkola (2007) discretize continuous data while learning the structure of a graphical model. Gat-Viks (2006) is another example of such joint inference.

In some applications, it would be reasonable to maintain ternary expression levels: 1 (upregulated), -1 (downregulated) and 0 (invariant). To address this issue, extensions to the Boolean logic-based models can be developed.

Specification of the Prior Distributions for the Model Parameters

The priors for the parameters θ_j are defined with Beta distribution, since in the model presented here they need to be constrained to the [0,1]-interval. We define the priors for the parameters θ_j with Beta distribution with hyperparameters a_j and b_j:

$\theta_j \sim Beta(a_j,b_j)$

The hyperparameters a_j and b_j are defined equal to 1, if γ_j =1: *Beta*(1,1). This makes the prior non-informative, allowing for the exploration of the whole parameter space. If γ_j =0, the proposal distributions for the pseudopriors can be calculated according to the method of Dellaportas. That is, the mean $mean_j$ and the variance var_j of the parameters θ_j are estimated from the pilot run of the saturated model, and the hyperparameters a_j and b_j are calculated by the formulas (*method of moments*):

$$a_j + b_j = \frac{mean_j(1 - mean_j)}{var_j} - 1,$$

$a_j=(a_j+b_j)mean_j$,

$b_j=(a_j+b_j)(1-mean_j)$

Next, one must define the prior distribution for the variables indicator γ. Since the terms γ_j are independent, each term can be specified with the independent Bernoulli distributions: $\gamma_j \sim Bernoulli(\pi_j)$, where π_j is the prior probability to include term *j* into the model. The simplest choice in variable selection problems is the uniform prior on γ, assuming that models are a priori equally probable, i.e. $\pi_j=\pi=0.5$. This prior is non-informative in the sense that it favours all models equally, but it is not non-informative with respect to the model size. If *p* is the number of potential regulators and *n* is the number of actual regulators, then $E(n)=0.5p$ and $var(n)=0.25p$ (Kohn *et al.*, 2001). This can be crucial for models with a sparse number of regulators, e.g. "AND" models with few gene regulators combined with AND-function, since the sampling procedure will not sample them at all. On the other hand, in case of models with high numbers of variables, we would like to favour more parsimonious models. It is advisable to set the probability π in a way to restrict *na priori* to lie in a short range. By setting $E(n)$ and $var(n)$ to the desired values, π can be calculated from:

$E(n)=\pi^*p, var(n)=\pi(1-\pi)p$.

A more flexible approach is to place a hyperprior on π:

$\pi \sim Beta(\alpha,\beta)$,

then the prior for the number of actual regulators *n* is Beta-binomial:

$n \sim Betabin(p,\alpha,\beta)$,

The values for α and β can be chosen by setting $E(n)$ and $var(n)$ to the desired values and solving the following equations (Kohn *et al.* 2001):

$$p\frac{\alpha}{\alpha+\beta} = E(n)$$

$$\frac{\alpha+1}{\alpha+\beta+1} = \frac{var(n)-E(n)(1-E(n))}{(p-1)E(n)}$$

While performing Gibbs variable selection with the complex models like "OR-NOR", we consider the same set of variables (genes) as potential activators and potential inhibitors. We use two indicators: γ^{act} and γ^{inh}, representing that a particular variable is included in the model as activator or inhibitor, respectively. To ensure that terms γ_j^{act} and γ_j^{inh} cannot be 1 at the same time, we specify γ_j^{inh} as:

$$\gamma_j^{inh} \sim Bernoulli((1-\gamma_j^{act})\pi_j^{inh}),$$

where π_j^{inh} is the prior probability for including the term *j* into the set of "true" inhibitors.

Implementation

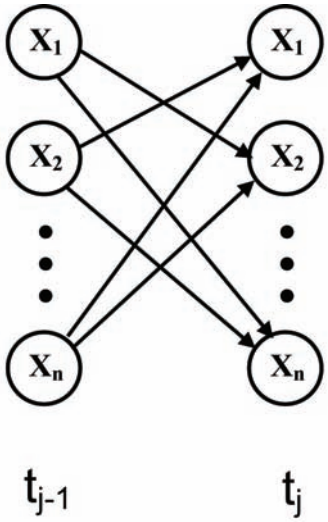
The Gibbs variable selection procedure described in this chapter is easily implemented by using BUGS (Bayesian Updating with Gibbs Sampling) which is the general purpose software for Gibbs sampling on graphical models (Thomas 2006, Spiegelhalter *et al*. 1996; Gilks 1993; Ntzoufras 1999). BUGS provides a declarative language for specifying a graphical model, and performs MCMC sampling from the resulting full conditional distributions. The system recognizes conditional conjugacy and uses it to sample efficiently. Failing that, it uses rejection and adaptive rejection methods or the Metropolis-Hastings algorithms. BUGS allows for the specification of a variety of prior distributions. There is a Windows version of BUGS, called WinBUGS, while the OpenBUGS software can be used on Unix-like platforms.

The BUGS output – samples of the MCMC chain – must be monitored for diagnosing slow convergence or lack of convergence. This can be done by using the package CODA implemented in **R** language (Plummer, 2006) or with a similar software **BOA** (Smith, 2007), see http://cran.r-project.org. CODA stands for Convergence Diagnostics and Output Analysis and BOA stands for Bayesian Output Analysis.

The BUGS code for our "OR" and "OR-NOR" models is presented in the Appendix.

The output of Markov chain simulation can be used to summarize the posterior distribution of the variables of interest: θ_j and γ_j. After the burn-in time, Markov chain samples are used to count the number of times γ_j had the value 1 in the Markov chain. For example, if the frequency of 1s exceeds 0.7, we assume that γ_j=1 and the respective regulator should be included in the "true" model. Otherwise, the regulator *j* should be excluded. The number of iterations for the burn-in time and for the estimations

Figure 5. Time-delay' gene regulation



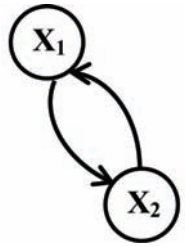
depends on the problem size. For moderate size problems, 10000 iterations for the burn-in and 10000 iterations for estimations will be probably sufficient.

As proposed by Gelman and Rubin (1992), a number of parallel runs of Markov chains should be carried out from different starting points. Convergence is diagnosed when the output from different Markov chains is indistinguishable. For parallel runs of Markov chains we use different initial values of the parameters indicator γ (when γ_j=0 for all j and when γ_j=0 for all j).

Model Checking

After the execution of the Gibbs variable selection and the estimation of the variables indicator γ, the check of goodness-of-fit of the model to data must be performed. Bayesian model checking uses the *posterior predictive distributions* (Gelman, 2000). The goal is to perform posterior predictions under the model and to assess the discrepancy between predicted and observed data. If the model is reasonably accurate, the predicted data should be similar to the observed data. Let γ be the observed data on Y and θ be the vector of parameters. Denote y^{rep} the *replicated* data generated under the model with the parameters θ. The posterior predictive distribution is

Figure 6. Feedback regulation among two genes



$$p(y^{rep} \mid y) = \int p(y^{rep} \mid \theta) p(\theta \mid y) d\theta.$$

The posterior predictive distribution can be computed by simulation: simulate parameters θ from their posterior distribution, and simulate y^{rep} from the sampling distribution $p(y^{rep}|\theta)$ conditioning on values of the simulated parameters. An advantage of using BUGS is that the generation of the replicate data can be easily incorporated into the model inference procedure.

Here, we wish to check the ability of the inferred regulatory model to predict the state of the gene Y from the states of its regulators. The inferred model is defined by the previously estimated binary vector γ, so the model contains now only the parameters θ. At each iteration of the MCMC, we generate the *replicate* data set $\{y^{rep}\}$ under the model based on the current simulated parameters θ.

Our model checking strategy is based on examining individual observations of Y $y_i, i=1,\ldots,N$ (N is the number of data samples) and comparing them to the replicate data. For the comparison we use the residual function $r_i=|y_i-E(y_i)|$, where the expectation $E(y_i)$ is estimated from the replicate dataset. Observations for which the residual is not close to 0 indicate some lack-of-fit of the model and should be regarded as an outlier. We regarded the residual as not close to 0 if its absolute value exceeded one estimated $var(y_i)$. The model prediction accuracy is calculated as the percentage of non-outliers.

Modeling Gene Expression Dynamics and Regulatory Feedback

Generally, there exist two kinds of microarray experiments: (1) measurements under different biological conditions e.g. in tumor and in normal tissues, and (2) time-series gene expression data. Bayesian network approaches are *static* in the sense that they represent causal relationships between variables at one point in time. They do not address the dynamic changes of the variables. This is particularly applicable in the first case, while the measurements of genes at each biological condition are treated as statistical samples. In case of time course data, two different regulatory situations can be considered. First, the state of gene *i* in a sample *j* depends on the states of its regulators in the same sample *j* ('simultaneous' regulation). Second, the state of the gene *i* in the sample *j* depends on the states of its regulators in the previous sample *j*-1 ('time delay' regulation), see Figure 5. Both situations can be treated by the method presented here. The time delay setting resembles the 'unrolled' Bayesian network i.e. the *Dynamic Bayesian network* model (Murphy, 2002). Inferring genetic regulatory networks with the Dynamic Bayesian models was presented e.g. by Perrin (2003), and is treated in this book. However, such approaches reconstruct only time-invariant regulatory influences, where the parameters are independent of time. The real dynamics of the genetic regulation can be resolved only when the parameters of the regulatory models will be allowed to vary in time. The problem of learning such models will then represent a great challenge in view of the lack of statistical data. Again, MCMC simulation techniques will come into play, which is the matter of future research.

Note that the 'unrolled' dynamic model makes it possible to infer feedback regulations, such as presented in Figure 6. Feedback relations between genes is a common motif in gene regulatory networks, identifying them is of great interest.

APPLICATION

The method presented in this chapter was tested with the gene expression data of the *S.cerevisiae* cell cycle (Bulashevska & Eils, 2005) and with the data on malaria parasite *Plasmodium falciparum* (Bulashevska et al., 2007). After the execution of the Bayesian model inference of the Boolean logic-based models and after the model checking, the results are being summarized in graphs. There, the full arcs represent activatory regulation and the dashed arcs represent inhibitory regulation. Genes pointing to one gene represent its regulators combined with the Boolean logic function underlying the model.

One example of the inferred "OR-NOR" regulatory interactions of *Plasmodium falciparum* in glycolysis is displayed in Figure 8. This is the result of the 'time delay' learning of the "OR-NOR"-model. The bloodborne pathogen *P.falciparum* causes the most fatal and prevalent form of malaria. Understanding the gene regulatory circuitry of this organism is of great importance. Since glycolysis is a crucial pathway in the maintenance of the parasite, we looked closely at the group of genes involved in it. From the public database PlasmoDB (http://plasmodb.org), twenty genes we harvested that are known to be involved in the glycolytic pathway. Many of the genes encode enzymes. Eighteen of the twenty genes were found in the dataset of Bozdech et al. (2003), which is the time-series gene expression data of the intraerythrocytic development cycle of *P.falciparum*. So the gene expression was measured every six hours at subsequent time points $t=1,\ldots,T$, where $T=53$. The true biological time resolution of the gene transcription and activation is yet unknown. In case of the 'time delay' learning, it is assumed that a gene becomes active at a subsequent time point after its regulators are active.

The deduced regulatory network (Figure 8) suggested the strategic position and hence the key regulatory role of the genes PF11_0157, PF11_0208, PF14_0341 and PF10_0155. Interestingly, the gene PF10_0155 was connected to both enzyme genes PF14_0341 and PF13_0141. The network revealed the groups of closely connected genes. One group contained the genes PF11_0157, PF13_0269, PF11_0294, PF14_0425 and PFC0831w, another: PF14_0341, PF10_0155, PF14_0598, PF11105w, etc. Bidirectional regulations e.g. between genes PF11_0157 and PF11_0294 might indicate that the genes are both active over long period of time and the proper arc direction could not be resolved. Another possibility is that both genes oscillate in a shifted manner. Feedback regulation through unmeasured biological mechanisms could also be hypothesized. The inhibitory connections between the genes suggest that the groups of genes work in a separated manner. The activation of the gene PF13_0269 by the gene PF11_0157 was shown previously experimentally. The metabolic pathway maps with enzymes for the *P.falciparum* glycolysis pathway, available at KEGG database, supported the predicted interactions. The predicted network provided more information than contained in the KEGG, though.

Obviously, different Boolean logic models have different semantics. For example, the "NOR"-model can suggest more inhibitors than the "OR-NOR"-model. Learning the "NOR"-model identifies only the inhibitors of a gene, i.e. the model "explains" the non-activity of the gene with the activity of its inhibitors. By the "OR-NOR"-model, the non-activity of the regulatee can also be "explained" with the failure of its activators. Generally, the "OR-NOR"-model gives valuable hypotheses on the most likely possible activators and inhibitors of each gene in the dataset. On the other hand, the "AND"-model is capable to reverse engineer the real synergistic relations between the genes, which is not possible by other approaches.

Although we have tested our approach on relatively moderate subsets of genes, the method can be readily applied to large datasets, where the advantages of the Bayesian variable selection arise.

CONCLUSION

This chapter has presented the models of genetic regulatory interactions possessing Boolean-logic semantics. They were formulated as probabilistic graphical models and placed into the context of the Bayesian modelling. In fact, they resemble the local interactions of nodes in the Bayesian network, though constrained and not combinatorial.

The modelling approach does not make an attempt to reconstruct the whole genetic regulatory network in one computational run, unlikely to the Bayesian network. Rather, the method is applied for each gene in the dataset, considering all other genes as candidate regulators, and then summarizing the results in a graph.

Bayesian modelling has a number of advantages. It allows for flexibility in defining complex models with many parameters. For example, by inserting into the model a new parameter, the variables indicator, we have converted the problem of model selection into the variable selection task, which is conveniently solved with Gibbs sampling. Generally, Markov Chain Monte Carlo simulation techniques rapidly evolve to facilitate Bayesian statistical inference. A further advantage of the Bayesian approach is that it enables to include subjective prior information into the model. For example, we used the subjective prior specification to enforce the number of gene regulators to lie in the desired range. Potentially, one could define priors aiming to incorporate into the model learning previous biological knowledge.

In the computational framework presented in this chapter, a particular regulatory Boolean-logic function (e.g. "AND", "AND-NOR", "OR-NOR" etc.) can be defined explicitly and the regulatory model can be learned from data. Given expression data on a gene and its potential regulators, the method permits to detect the most likely regulators of the gene. The main advantage of the present approach is that the elucidated gene relationships do not require laborious manual analysis for their interpretation, in contrast to the arbitrary combinatorial interactions learned by means of standard Bayesian networks models. On the other hand, the method enables to reveal more complex multi-gene relations than those defined in the conventional regression models.

Generally, the Bayesian variable selection under the so called $n > p$ or 'large p, small n' paradigm, when the sample size n is substantially smaller than the number of covariates in the regression, remains an important point of statistical research. The problem of selecting significant gene regulators based on microarray data apparently represents such a 'large p, small n' problem. In West (2003), the number of covariates was projected to lower dimension using principal component. Bayesian variable selection that introduces sparseness through priors on the model size and on the role of each individual gene is a powerful approach, well suited to the problem of reconstructing the genetic regulatory network.

RELATED APPROACHES

Probabilistic Graphical Models for Cellular Networks

Probabilistic graphical models have become an important tool for computational analysis of biological data.

The system MinReg (Pe'er et al. 2002) was designed with the same goal as discussed here to constrain Bayesian networks to parsimonious models, in order to make them more biologically relevant. The idea is that biological regulatory networks have a limited number of "master regulators", which affect the

transcription of large numbers of genes. The authors constrain the number of regulators of each gene and the total number of regulators in the model. A regulator is then reliable when it regulates a whole set of target genes. The authors developed an iterative algorithm for searching for high scoring networks, while using the Bayesian score for local models. A relationship not identifiable by the MinReg is the cooperative activity of regulators ("AND"-model). With this respect, our approach is of advantage.

The *Module Networks* (Segal et al. 2003, 2005) is a further probabilistic framework. The system assigns genes into modules. Each module is regulated by a *regulation program* that is a set of rules organized as a regression tree. Expectation-Maximization algorithm (EM) was developed to iteratively search for models with the highest Bayesian score.

Gat-Viks et al. (2006) use a probabilistic *factor graph* model to jointly model continuous high-throughput experimental data and a priori known regulatory relations. A factor graph is a bipartite graph associating variable nodes with factor nodes. The variable nodes are used both to represent continuous experimental data and the respective discrete states of the genes. Also, two kinds of factor nodes are used: one for *discretizer distributions* (mixtures of Gaussians), which specify the joint distribution of the discrete states and the continuous observations, the other one - for the regulatory functions, which are the Bayesian network's distributions. Given the model, predictions are made with *Loopy Belief Propagation*. The predictions are then compared to the experimental measurements; in case discrepancy is found, the model is iteratively refined. Given a target gene and its candidate regulatory unit the refinement process searches in the space of regulatory functions to achieve the best Bayesian score. During the refinement, the discretization parameters are re-optimized with the Expectation-Maximization (EM) procedure. This modelling approach is implemented in the software tool MetaReg (Ulitsky, 2008).

Recent efforts are dedicated to the integration of the gene expression data with other biological sources, such as promoter sequences, *cis*-elements, ChIP-chip data etc. (Bussemaker et al., 2007; Beer and Tavazoie, 2004; Hartemink et al., 2002; Segal et al., 2002; Bar-Joseph et al., 2003).

Further Applications of the Bayesian Variable Selection in Genomics

Bayesian variable selection, applied in this chapter for the elucidation of regulatory interactions between genes, is also being adopted, however, in a supervised problem, where the goal is to select a subset of genes/markers that are more influential than the others for classification of cancer phenotypes, disease stages etc. In this context, probit or logistic regression models are applied based on the seminal paper of Albert & Chib (1993). The authors proposed an auxiliary variable approach for binary probit regression model introducing latent variables in the model and rendering the conditional distributions of the model parameters to normal form. Albert & Chib used the *block Gibbs sampler*. Holmes & Held (2006) extended this approach using joint updating of the regression coefficients and the auxiliary variables, thus improving the performance. Besides, they adopted the auxiliary approach to logistic regression. With microarray data on breast tumors, Lee et al. (2003) used probit regression model relating continuous gene expression levels to the binary response: patient is carrying mutations in BRCA1 or BRCA2 genes, or not. The variables indicator γ was introduced into the model, and the number of selected genes was restrained by choosing probability π of inclusion of a gene into the model to be small, as already pointed out in this chapter. The prior for the regression coefficients β_γ was chosen as: $\beta_\gamma \sim N(0, c(X_\gamma' X_\gamma)^{-1})$ where c is a positive scale factor determining the degree of shrinkage of the coefficients through the posterior distributions. Smith and Kohn (1997) recommend choosing c between 10 and 100. Sha et al.

(2004) developed a Bayesian variable selection method for the multinomial probit model to identify molecular signatures of two disease stages of rheumatoid arthritis. They also discuss the choice of *c* (see Brown et al., 2002).

Since the regulatory models described in this chapter aim to mimick the cooperative binding of transcription factors to the promoter region of the regulated gene, they appeared to be similar to the models being applied to relate transcription factor binding sites (TFBS) to the expression of the respective genes. The goal of such settings is to find the TFBS with the strongest predictive power (predictive models of gene expression). Liu et al. (2006) use the transcription factors library TRANSFAC (Matys (2003), BIOBASE GmbH, see http://www.gene-regulation.com/pub/databases.html) to identify the TFBS candidates, and then employ linear regression model in case of continuous values of gene expression and probit regression model with the discrete expression levels. The authors perform the Bayesian variable selection with Gibbs sampling. Tadesse et al. (2004) use the similar setting of the Bayesian variable selection to identify DNA-binding sites (*regulatory motifs*) which explain the expression of genes by previously generating a large list of candidate motifs with MotifRegressor (Conlon, 2003).

Selection of Regressors

The framework presented in this chapter resembles the statistical problem of the selection of predictors in a regression setting. Clyde et al. (1996) pointed out that the correlation of predictors is a serious obstacle. Making explanatory variables orthogonal to each other can improve statistical learning, particularly convergence and mixing of MCMC.

Chipman (1996) discussed a strategy to reduce model space by grouping the predictors and to consider importance of the groups instead of individual variables in the regression.

Most of the works end up with the selection of main effects, ignoring the interaction effects of the predictors. Chen (2004) proposed a Bayesian variable selection method with a goal to elucidate interactions - BSI (*Bayesian Selection of Interactions*). It extends the framework of SSVS (George & McCulloch, 1993) and introduces priors for pair-wise interactions as well as joint priors to express the dependence of the main effects on the interactions. Bayesian model averaging by using a set of *a posteriori* likely models (Madigan and Raftery, 1994, Clyde 1999) can also be employed with the aim of variable selection (Brown, 2002).

FUTURE RESEARCH DIRECTIONS

Gene expression measurements represent high dimensional data with small number of sample cases. Elucidating complex dependencies from this data raises a great statistical challenge. For regression-like models with large numbers of candidate predictors (‘large *p*, small *n*’ problems), the Bayesian variable selection approach described in this chapter still remains a matter of current research. Slow mixing and bad convergence of the Markov chains is a major problem. MCMC algorithms like Gibbs sampling take more time wandering around less interesting regions of the model space, often remain stuck in local maxima and do not provide an adequate representation of the model space with the increasingly complex patterns of collinearity. In this context, future research will proceed in two directions. Firstly, with respect to the model specification, the formulation of the prior for the variables indicator γ will further evolve. Secondly, from the computational perspective, the sophisticated MCMC algorithms

will be designed capable to quickly and adequately explore the high-dimensional model space, and to identify regions of high posterior probability over models. For example, the *shotgun stochastic search* (SSS) approach was developed by Hans et al. (2005). It is inspired by the Metropolis-Hastings MCMC algorithm, but can more rapidly identify probable models by evaluating many neighbourhood models in parallel as proposals. Moreover, the parallel implementation of the method for use on a Unix-cluster was provided.

The development of trans-dimensional Markov Chain Monte Carlo, originating from the work of Green (1995), is an important future research direction. Jasra et al. (2007) proposed the *population-based reversible jump MCMC* which combines the advantages of both population-based and reversible jump approaches. The population-based simulation simultaneously represents many properties of the target distribution and can provide an improved dimension-changing jumping; whereas the standard reversible jump method does not retain information about which states have been visited and has greater capacity to discover new states.

Despite a substantial amount of works aiming to reveal functionally important genes that regulate other genes or are significantly predictive for classification of different biological phenotypes on macro levels, deducing the complex dependencies between the genes still remains a challenge. The model described in this chapter is a step in this direction. Introducing time evolution in the regulatory network models and considering the entire gene data in the global model is a highly challenging future perspective.

REFERENCES

Akutsu, T., Miyano, S., & Kuhara, S. (2000). Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics (Oxford, England), 16*, 727–734. doi:10.1093/bioinformatics/16.8.727

Albert, J., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association, 88*, 669–679. doi:10.2307/2290350

Bar-Joseph, Z., Gerber, G. K., Lee, T. I., Rinaldi, N. J., Yoo, J. Y., & Robert, F. (2003). Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, *21*(11), 1337–1342. doi:10.1038/nbt890

Bedrick, E. J., Christensen, R., & Johnson, W. (1996). A new perspective on priors for generalized linear models . *Journal of the American Statistical Association*, *91*, 1450–1461. doi:10.2307/2291571

Beer, M. A., & Tavazoie, S. (2004). Predicting gene expression from sequence. *Cell*, 16, *117*(2), 185-98.

Bernardo, J. M., & Smith, A. F. M. (1994). *Bayesian theory. Wiley series in probability and mathematical statistics*. Chichester: John Wiley and Sons.

Bozdech, Z., Llinás, M., Pulliam, B. L., Wong, E. D., Zhu, J., & DeRisi, J. L. (2003). The transcriptome of the intraerythrocytic development cycle of plasmodium falciparum . *PLoS Biology*, *1*(1), E5. doi:10.1371/journal.pbio.0000005

Brown, P. J., Vanucci, M., & Fearn, T. (2002). Bayes model averaging with selection of regressors. *Journal of the Royal Statistical Society. Series B. Methodological*, *64*, 519–536. doi:10.1111/1467-9868.00348

Bulashevska, S., Adebiyi, E., Brors, B., & Eils, R. (2007). New insights into the genetic regulation of Plasmodium falciparum obtained by Bayesian modeling. *Gene Regulation and Systems Biology*, *1*, 117–129.

Bulashevska, S., & Eils, R. (2005). Inferring genetic regulatory logic from expression data. *Bioinformatics (Oxford, England)*, *21*(11), 2706–2713. doi:10.1093/bioinformatics/bti388

Bussemaker, H. J., Foat, B. C., & Ward, L. D. (2007). Predictive modeling of genomewide mRNA expression: From modules to molecules. *Annual Review of Biophysics and Biomolecular Structure*, *36*, 329–347. doi:10.1146/annurev.biophys.36.040306.132725

Carlin, B. P., & Chib, S. (1995). Bayesian model choice via Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B. Methodological*, *57*, 473–484.

Chen, M.-H., & Ibrahim, J. G. (2003). Conjugate priors for generalized linear models. *Statistica Sinica*, *13*, 461–476.

Chipman, H. (1996). Bayesian variable selection with related predictors. *The Canadian Journal of Statistics*, *24*, 17–36. doi:10.2307/3315687

Chung, T.-H., Brun, M., & Kim, S. (2006). Quantization of global gene expression data. *5th International Conference on Machine Learning and Applications* (ICMLA'06) (pp. 187–192).

Clyde, M. (1999). Bayesian model averaging and model search strategies (with discussion). In J. Bernardo, J. Berger, A. Dawid & A. Smith (Eds.), *Bayesian statistics*, 6. Oxford: Clarendon Press.

Clyde, M., DeSimone, H., & Parmigiani, G. (1996). Prediction via orthogonalized model mixing. *Journal of the American Statistical Association*, *91*, 1197–1208. doi:10.2307/2291738

Conlon, E. M., Liu, X. S., Lieb, J. D., & Liu, J. S. (2003). Integrating regulatory motif discovery and genomewide expression analysis. *Proceedings of the National Academy of Sciences of the United States of America*, *100*, 3339–3344. doi:10.1073/pnas.0630591100

Cooper, G. F., & Herskovits, E. H. (1992). The induction of probabilistic networks from data. *Machine Learning*, *9*(4), 309–347.

Dellaportas, P., Forster, J. J., & Ntzoufras, I. (2000). *Bayesian variable selection using the Gibbs sampler*. In D. K. Dey, S. Ghosh & B. Mallick (Eds.), *Generalized linear models: A Bayesian perspective* (pp. 271-286). New York: Marcel Dekker.

Dellaportas, P., Forster, J. J., & Ntzoufras, I. (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing*, *12*, 27–36. doi:10.1023/A:1013164120801

Di Camillo, B., Sanchez-Cabo, F., Toffolo, G., Nair, S. K., Trajanoski, Z., & Cobelli, C. (2005, December 1). A quantization method based on threshold optimization for microarray short time series. *BMC Bioinformatics*, *6*(Suppl 4), S11. doi:10.1186/1471-2105-6-S4-S11

Friedman, N., Linial, M., Nachman, I., & Peer, D. (2000). Using Bayesian network to analyze expression data. *Journal of Computational Biology*, *7*, 601–620. doi:10.1089/106652700750050961

Gamerman, D., & Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic simulation for Bayesian inference*. Chapman & Hall/CRC.

Gat-Viks, I., Tanay, A., Raijman, D., & Shamir, R. (2006). A probabilistic methodology for integrating knowledge and experiments on biological networks. *Journal of Computational Biology*, *13*(2), 165–181. doi:10.1089/cmb.2006.13.165

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2000). *Bayesian data analysis*. Chapman & Hall/CRC.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–472. doi:10.1214/ss/1177011136

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741. doi:10.1109/TPAMI.1984.4767596

George, E. I., & McCulloch, R. E. (1993). Stochastic search variable selection. In W. R. Gilks, S. Richardson & D. J. Spiegelhalter (Eds.), *Markov Chain Monte Carlo in Practice* (pp. 203-214). London: Chapman and Hall.

Gersho, A., & Gray, R.,M. (1992). *Vector quantization and signal compression*. The Kluwer International Series in Engineering and Computer Science.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1993). *Markov Chain Monte Carlo in practice*. London: Chapman & Hall.

Godsill, S. J. (2001). On the relationship between Markov Chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, *10*, 1–19. doi:10.1198/10618600152627924

Green, P. J. (1995). Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika*, *82*, 711–732. doi:10.1093/biomet/82.4.711

Hans, C., Dobra, A., & West, M. (2007). Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association*, *102*(478), 507–516. doi:10.1198/016214507000000121

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2002). Combining location and expression data for principled discovery of genetic regulatory network models. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 437–449.

Heckerman, D. (1998) *A tutorial on learning with Bayesian networks*. In M. I. Jordan (Ed.), *Learning in graphical models*. Dordrecht: Kluwer Academic Publishers.

Heckerman, D., & Breese, J. S. (1994). Causal independence for probability assessment and inference using Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, *6*, 826–831.

Holmes, C. C., & Held, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis (Online)*, *1*, 55–67.

Jaakkola, T. S. (1997). *Variational methods for inference and estimation in graphical models*. Retrieved from ftp://ftp.ai.mit.edu/pub/users/tommi/thesis.ps.gz

Jasra, A., Stephens, D. A., & Holmes, C. C. (2007). Population-based reversible jump Markov Chain Monte Carlo. *Biometrika, 19*, 1–21.

Jensen, F. V. (1996). *Introduction to Bayesian metworks*. New York: Springer.

Jordan, M. I. (2004). Graphical models. *Statistical Science*, *19*, 140–155. doi:10.1214/088342304000000026

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*, 437–467. doi:10.1016/0022-5193(69)90015-0

Kauffman, S. A. (1996). *The origins of order: Self-organization and selection in evolution*. New York: Oxford University Press.

Kohn, R., Smith, M., & Chan, D. (2001). Nonparametric regression using linear combinations of basis functions. *Statistics and Computing*, *11*, 313–322. doi:10.1023/A:1011916902934

Kuo, L., & Mallick, B. (1998). Variable selection for regression models. *Sankhya*, B, 60, Part 1, 65-81.

Lee, K. E., Sha, N., Dougherty, E. R., Vannucci, M., & Mallick, B. K. (2003). Gene selection: A Bayesian variable selection approach. *Bioinformatics (Oxford, England)*, *19*(1), 90–97. doi:10.1093/bioinformatics/19.1.90

Liang, S., Fuhrman, S., & Somogyi, R. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Proc. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *3*, 18–29.

Liu, K.-Y., Zhou, X., Kan, K., & Wong, S. T. C. (2006). Bayesian variable selection for gene expression modeling with regulatory motif binding sites in neuroinflammatory events. *Neuroinformatics*, *6*, 95–117. doi:10.1385/NI:4:1:95

Madigan, D. M., & Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's Window. *Journal of the American Statistical Association*, *89*, 1335–1346. doi:10.2307/2291017

Matys, V., Fricke, E., & Geffers, R. (2003). TRANSFAC: Transcriptional regulation, from patterns to profiles. *Nucleic Acids Research, 31*, 374–378. doi:10.1093/nar/gkg108

McCullagh, P., & Nelder, J. A. (1983). *Generalized linear models*. London: Chapman & Hall.

Meek, C., & Heckerman, D. (1997). Structure and parameter learning for causal independence and causal interaction models. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence* (pp. 366-375).

Murphy, K. P. (2002). *Dynamic Bayesian networks*: *Representation, inference, and learning*. Unpublished doctoral thesis, UC Berkeley.

Ntzoufras, I. (1999). *Gibbs variable selection using BUGS* (Tech. Rep.). Retrieved from http://www.ba.aegean.gr/ntzoufras/tr.htm

Pe'er, D., Regev, A., & Tanay, A. (2002). Minreg: Inferring an active regulator set. *Bioinformatics (Oxford, England), 18*(Suppl. 1), 258–267.

Pe'er, D., Tanay, A., & Regev, A. (2006). MinReg: A scalable algorithm for learning parsimonious regulatory networks in yeast and mammals. *Journal of Machine Learning Research, 7*, 167–189.

Pearl, J. (1987). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence, 32*, 245–257. doi:10.1016/0004-3702(87)90012-9

Pearl, J. (1998). *Probabilistic reasoning in intelligent systems*. San Francisco: Morgan Kaufmann.

Perrin, B. E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., & d'Alché-Buc, F. (2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics (Oxford, England), 19*(Suppl. 2), 38–48. doi:10.1093/bioinformatics/btg1071

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC. *R News, 6*(1), 7-11. Retrieved from http://CRAN.R-project.org/doc/Rnews/

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*(2), 461–464. doi:10.1214/aos/1176344136

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., & Friedman, N. (2003). Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics, 34*(2), 166–176. doi:10.1038/ng1165

Segal, E., Wang, H., & Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics (Oxford, England), 19*(Suppl. 1), i264–i272. doi:10.1093/bioinformatics/btg1037

Segal, E., Yelensky, R., & Koller, D. (2003b). Genomewide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics (Oxford, England), 19*(Suppl. 1), 273–282. doi:10.1093/bioinformatics/btg1038

Sha, N., & Vanucci, M. (2004). Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics, 60*, 812–819. doi:10.1111/j.0006-341X.2004.00233.x

Shmulevich, I., Dougherty, E. R., Seungchan, K., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics (Oxford, England), 18*, 261–274. doi:10.1093/bioinformatics/18.2.261

Smith, B. J. (2007). Boa: An r package for MCMC output convergence assessment and posterior inference. *Journal of Statistical Software, 21*(11).

Smith, M., & Kohn, R. (1997). A Bayesian approach to nonparametric bivariate regression. *Journal of the American Statistical Association, 92*, 1522–1535. doi:10.2307/2965423

Somogyi, R., & Sniegosky, C. A. (1996). Modeling the complexity of genetic networks: Understanding multigenic and pleiotropic regulation. *Complexity*, 45.

Spiegelhalter, D. J., Thomas, A., & Best, N. G. (1996). Computation on Bayesian graphical models. In J. M. Bernardo, J. O. Berger, A. P. Dawid & A. F. M. Smith (Eds.), *Bayesian statistics*, 5, 407-425.

Srinivas, S. (1993). A generalization of the noisy-or model. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence.*

Steck, H., & Jaakkola, T. (2007). Predictive discretization during model selection. In *Proc. 11th International Conference on Artificial Intelligence and Statistics*.

Tadesse, M. G., Vanucci, M., & Lio, P. (2004). Identification of DNA regulatory motifs using Bayesian variable selection. *Bioinformatics (Oxford, England)*, *2*(16), 2553–2561. doi:10.1093/bioinformatics/bth282

Thomas, A., O'Hara, B., Ligges, U., & Sturtz, S. (2006). Making BUGS open. *R News*, *6*, 12–17.

Ulitsky, I., Gat-Viks, I., & Shamir, R. (2008). MetaReg: A platform for modeling, analysis, and visualization of biological systems using large-scale experimental data. *Genome Biol*., 2; 9(1): R1.

West, M. (2003). Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics*, *7*, 723–732.

ADDITIONAL READING

Andrieu, C., Djuric, P. M., & Doucet, A. (2001). Model selection by MCMC computation. *Signal Processing*, *81*, 19–37. doi:10.1016/S0165-1684(00)00188-2

Carlin, B. P., & Louis, T. A. (2000). *Bayes and empirical bayes methods for data analysis*. Chapman & Hall/CRC.

Doucet, A. (2001). *Sequential Monte Carlo methods in practice*. *Statistics for engineering and information science*. Springer.

Fishman, G. S. (2006). *A first course in Monte Carlo*. Southbank, Australia: Thomson Brooks.

Gelman, A. (2004). *Bayesian data analysis*, 2nd ed. Chapman & Hall/CRC.

Gelman, A. (2007). *Data analysis using regression and multilevel hierarchical models*. Cambridge University Press.

Ghosh, J. K., Delampady, M., & Samanta, T. (2006). *An introduction to Bayesian analysis: Theory and methods*. Springer Texts in Statistics.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*. Springer series in statistics.

Marin, J.-M., & Robert, C. P. (2007). *Bayesian core: A practical approach to computational Bayesian statistics*. New York: Springer.

Mengersen, K. L., Robert, C. P., & Guhenneuc-Jouyaux, C. (1999). MCMC convergence diagnostics: A review. *Bayesian Statistics, 6*, 415–440.

Neapolitan, R. E. (2003). *Learning Bayesian network*. Prentice Hall.

Pan, W. (2005). Incorporating biological information as a prior in an empirical Bayes approach to analyzing microarray data. *Statistical Applications in Genetics and Molecular Biology, 4*, 12. doi:10.2202/1544-6115.1124

Pournara, I., & Wernisch, L. (2007). Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC Bioinformatics, 8*, 61. doi:10.1186/1471-2105-8-61

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods*, 2nd ed. Springer texts in statistics. New York: Springer.

Rogers, S., Khanin, R., & Girolami M. (2006). Bayesian model-based inference of transcription factor activity. *BMC Bioinformatics, 8*(Suppl 2).

Rubinstein, R. Y., & Kroese, D. P. (2007). *Simulation and the Monte Carlo Method*. Wiley Series in Probability and Statistics, Wiley & Sons.

Sha, N., Tadesse, M. G., & Vannucci, M. (2006). Bayesian variable selection for the analysis of microarray data with censored outcomes. *Bioinformatics (Oxford, England), 22*(18), 2262–2268. doi:10.1093/bioinformatics/btl362

Sisson, S. A. (2005). Transdimensional Markov chains: A decade of progress and future perspectives. *Journal of the American Statistical Association*, online.

KEY TERMS AND DEFINITIONS

Bayesian Inference: A statistical inference method in which the degree of belief in a hypothesis is expressed in terms of probability distributions *a priori* i.e. before evidence has been observed, and is updated using evidence with the help of the Bayes' theorem.

Bayesian Network: A probabilistic graphical model representing conditional independencies of random variables via a directed acyclic graph (DAG). A Bayesian network is specified by a graph structure and conditional probability distributions (CPDs) for each node, conditional upon its parents in the graph. Algorithms exist that perform inference and learning in Bayesian networks.

Bayesian Variable Selection: A problem of identifying a subset of predictors from a large set of potential predictors in the regression-like models. Bayesian approach is promising due to efficient *a priori* parameter formulations.

Boolean Network: A set of Boolean variables connected in the network, where the state of each variable is determined by the states of its neighbours by Boolean functions.

Genetic Regulatory Network: An abstract representation of the orchestrated regulation of expression of genes.

Gibbs Sampling: Is a special case of the MCMC sampling algorithms named after the physicist J. W. Gibbs. The algorithm samples from the joint probability distribution of random variables by generating an instance from the distribution of each variable in turn, conditional on the current values of the other variables.

Graphical Models: Graphs with nodes representing random variables, where arcs encode conditional independencies between the variables.

Markov Chain Monte Carlo (MCMC): A class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its equilibrium distribution.

Noisy-OR Model: A special case of the specification of the CPD in the Bayesian network, where the number of parameters is linear on the number of parents of a node. The idea is that each parent is capable to execute its influence on the node independently of other parents, whereby the individual effects are then summarized with the Boolean function OR.

Probabilistic Modeling: A kind of modelling where a problem space is expressed in terms of random variables and their probability distributions. Properties of the underlying distributions are being deduced from data in the process of probabilistic inference.

APPENDIX

BUGS code for the “OR”-model

```
model OR-model {
   # specification of the likelihood
   for (i in 1:N){
      for (j in 1:P){
           s[i,j] ~ dbern(theta[j])
           I[i,j] <- X[i,j]*s[i,j]*gamma[j]
      }
      sum[i] <- sum(I[i, ])
      constraint[i] <- step(sum[i]-1)
      Y[i] ~ dbern(constraint[i])
      # Y[i]=1 if sum[i]≥1 i.e. at least one of I[i,j]=1
# gene Y in sample i is active, if one of its activators is
# active
   }
   # specification of the priors
   for (j in 1:P){
      gamma[j] ~ dbern(pi)
      pi <- 0.2
      theta[j] ~ dbeta(a[j],b[j])
      a[j] <- 1
      b[j] <- 1
      # alternatively: hyperprior on pi
      # gamma[j] ~ dbern(pi[j])
      # pi[j] ~ dbeta(api[j],bpi[j])
      # api[j] <- …
      # bpi[j] <- …
      # values to keep the number of regulators in the desired range

      # method of Dellaportas:
      # if gamma[j]=0 use proposal values aprop[j] and bprop[j]
# calculated based on mean and variance estimations of theta
      # from the pilot run of the saturated model
      # if gamma[j]=1 a[j]=1, b[j]=1 (non-informative prior)

      # gamma[j]<-1.0 for sampling from saturated model
      # theta[j] ~ dbeta(a[j],b[j])
      # a[j] <- gamma[j] + (1-gamma[j])*aprop[j]
           # b[j] <- gamma[j] + (1-gamma[j])*bprop[j]
# aprop[j]<- (priormean[j]*(1-priormean[j])/pow(priorvar[j],2)-
```

```
1)*priormean[j]
# bprop[j]<- (priormean[j]*(1-priormean[j])/pow(priorvar[j],2)-
1)*(1-priormean[j])
        }
 }
```

BUGS code for the “OR-NOR”-model

```
model OR-NOR-model
{
   for (i in 1:N){
      for (j in 1:P){
           s[i,j] ~ dbern(theta[j])
           I[i,j] <- X[i,j]*s[i,j]
           Iact[i,j] <- I[i,j]*gamma_act[j]
           Iinh[i,j] <- 1-I[i,j] + I[i,j]*(1-gamma_inh[j])
      }
      sumact[i] <- sum(Iact[i, ])
      constraint_act[i] <- step(sumact[i]-1)
      suminh[i] <- sum(Iinh[i, ])
      constraint_inh[i] <- step(suminh[i]-P)
      constraint[i] <- constraint_act[i] * constraint_inh[i]
      Y[i] ~ dbern(constraint[i])
      # gene Y in sample i is active, if one of the activators
      # is active and all inhibitors are not active
   }
   for (j in 1:P){
      gamma_act[j] ~ dbern(pi)
      pinh[j] <- (1-gamma_act[j])*pi
      gamma_inh[j] ~ dbern(pinh[j])
      # if gamma_act[j]=1, pinh[j]=0 (gene j is already activator)
      pi <- 0.2
      # alternatively: see above

      theta[j] ~ dbeta(a[j],b[j])
      a[j] <- 1
      b[j] <- 1
      # method of Dellaportas
      # pseudopriors if gene j is neither activator nor inhibitor
      a[j] <- 1 + (1-gamma_act[j])*(1-gamma_inh[j])*aprop[j]
      b[j] <-  1 + (1-gamma_act[j])*(1-gamma_inh[j])*bprop[j]
      # use proposal values aprop[j] and bprop[j]
```

```
   # calculated based on mean and variance estimations of theta
      # from the pilot run of the saturated model
      # gamma_act[j]<-1.0
aprop[j]<- (priormean[j]*(1-priormean[j])/pow(priorvar[j],2)-1)*
priormean[j]
bprop[j]<- (priormean[j]*(1-priormean[j])/pow(priorvar[j],2)-1)*(1-
priormean[j])
        }
 }
```

Chapter 6
A Bayes Regularized Ordinary Differential Equation Model for the Inference of Gene Regulatory Networks

Nicole Radde
University of Leipzig, Germany

Lars Kaderali
University of Heidelberg, Germany

ABSTRACT

Differential equation models provide a detailed, quantitative description of transcription regulatory networks. However, due to the large number of model parameters, they are usually applicable to small networks only, with at most a few dozen genes. Moreover, they are not well suited to deal with noisy data. In this chapter, we show how to circumvent these limitations by integrating an ordinary differential equation model into a stochastic framework. The resulting model is then embedded into a Bayesian learning approach. We integrate the-biologically motivated-expectation of sparse connectivity in the network into the inference process using a specifically defined prior distribution on model parameters. The approach is evaluated on simulated data and a dataset of the transcriptional network governing the yeast cell cycle.

INTRODUCTION

Developments in experimental technologies such as DNA microarrays and real-time PCR experiments render quantitative measurements of expression levels of a large number of genes feasible, and make the acquisition of time series concentration data possible. Such data can be used to reconstruct gene regulatory networks from the data, and to derive detailed quantitative models describing the dynamics of a system under consideration. These models can then be used to run simulations, to study the effect of particular interventions, and to analyze the dynamic behavior of the network under various conditions.

DOI: 10.4018/978-1-60566-685-3.ch006

Several different approaches have been developed in the last decade to infer regulatory networks from gene expression measurements. These approaches differ in the level of detail used to describe regulatory control mechanisms, and in the methods employed to estimate model parameters. The most frequent models used are correlation based models, models based on information theory, Boolean networks, Bayesian networks or, more generally, graphical models, and ordinary differential equations. Our focus in the following will be on the latter, and we will show how to integrate them into a probabilistic framework, which allows it to apply a Bayesian learning approach to parameter estimation.

Ordinary differential equations provide a quantitative time and state continuous description of a system's dynamic behavior. They are usually based on chemical reaction kinetics, and model parameters correspond directly to reaction rates, binding affinities and degradation rates. Therefore, they provide a very detailed and realistic description of a system under consideration. On the downside, the consequence of this detailed description in view of limited data is that the number of model parameters to be estimated usually far exceeds the number of measurements available. Parameter estimation then leads to underdetermined optimization problems. It is for this reason that in practice, network inference with (nonlinear) differential equation models is limited to networks of at most a few dozen components.

Another disadvantage of ordinary differential equation models is that they are not well suited to handle noisy data. However, experimental data are often prone to considerable noise. This further complicates the estimation of model parameters, since learning algorithms may simply tune to the noise in the data, instead of deriving true biological mechanisms.

In the following, we will describe an inference approach for gene regulatory networks from time series gene expression data which combines the detailed quantitative dynamics of differential equation models with a probabilistic modeling approach, thus taking noisy measurements into account. Parameters in this framework are estimated using Bayes' theorem. The problem with underdetermined models can then be addressed by integrating additional assumptions on model parameters through suitably chosen prior distributions. We discuss one particular prior distribution, which drives the inference to sparse networks. We then show that this enables the method to cope with datasets consisting of only few time points and a large number of model parameters. This makes the method particularly suitable for the task of quantitative modeling from typical real-world experimental datasets. We illustrate this claim both on simulated and real gene expression data from the transcriptional network governing the yeast cell cycle. Finally, we discuss relations between Bayes regularized differential equation models and other stochastic approaches from a more general point of view.

BACKGROUND

We will now derive the system of differential equations we use to model genetic regulatory networks. The underlying assumption is that these equations describe the true states of the biological system, which is hence a deterministic system. We will discuss this assumption and its consequences in more detail later.

Differential Equation Models for Gene Regulatory Networks

Ordinary differential equations (ODE) offer a deterministic, time and state continuous description of a system's temporal evolution. In these models, a gene regulatory network is understood as a system *S*, consisting of *n* interacting components. At any time *t*, *S* is assumed to be fully characterized by the *state* $x(t)=(x_1(t),x_2(t),...,x_n(t))\in\tilde{N}^n$, where variable $x_i(t)$ corresponds to the concentration of gene product *i* at time *t*. The state space W is usually the $\tilde{N}^n$, and time $t\in\tilde{N}$. The dynamic behavior of *S* is characterized by a function $\Phi:W'\tilde{N}\rightarrow W$, which assigns each tuple $(x,t)\in W'\tilde{N}$ an element in the state space W. The function $x(t)=\Phi(x_0,t)$ is assumed to be the solution of the initial value problem

$$\dot{x}(t) = f(x(t)), \quad x(t_0) = x_0 \tag{1}$$

with initial state x_0 and a continuously differentiable function f: $W\rightarrow\tilde{N}^n$.

Systems of differential equations have been used in recent years to model the dynamic behavior of gene regulatory networks quantitatively. A commonly used parameterization of the function *f* is given by

$$f_i(x(t)) = s_i - \gamma_i x_i(t) + g_i(x(t)), \quad i = 1,...,n. \tag{2}$$

The *basic synthesis rate* $s_i \geq 0$ describes the expression rate of gene *i* when no regulators of *i* are present. Degradation of gene product *i* is assumed to be a first order decay process. Hence degradation is proportional to the concentration of the gene product, with a *degradation rate* γ_i. Finally, the *regulation function* g_i accounts for influences of network components regulating the expression of gene *i*.

Isolating an initial amount $x_i(0)$ of gene product *i* at time $t = 0$, the molecules are degraded, and the dynamic of $x_i(t)$ is described by

$$\dot{x}(t) = -\gamma_i x_i(t), \quad x(0) = x_0. \tag{3}$$

A solution of this initial value problem is an exponentially decreasing function

$$x_i(t) = x_{i,0}e^{-\gamma_i t}, \tag{4}$$

characterized by its *half-life* $T_{½}$. $T_{½}$ denotes the time after which $x_i(t)$ has dropped to half of its initial value, $x_i(T_{½}) = x_i(0)/2$. Degradation rate and half-life are related via $T_{½} = \ln(2)/\gamma_i$.

The course of component *i* in the absence of any regulators is described by

$$\dot{x}(t) = s_i - \gamma_i x_i(t), \quad x(0) = x_0. \tag{5}$$

Starting with an initial value $x_i(0)$, the solution of this system exponentially approaches the steady state concentration $x_{i,s} = s_i/\gamma_i$. Thus, all genes which are not regulated by other genes in the network eventually reach a steady state.

The differential equations are coupled through the regulation functions $g_i(x(t))$. Usually, for the sake of simplicity, g_i is taken to be the sum of *individual regulation functions* $r_{ij}(x_j(t))$, assuming that the effects of different regulators can be described independently from one another, and the total effect of all regulators on variable x_i is the sum of these individual effects:

$$g_i(x(t)) = \sum_{j=1}^{n} r_{ij}(x_j(t)). \tag{6}$$

This independence assumption neglects processes such as complex formation or cooperative binding between different transcription factors. It can thus be crucial when these interactions play a dominant role in gene expression regulation. On the other hand, an inclusion of all possible cooperative effects would lead to a far more complex model, and the additivity assumption can be seen as a trade-off between tractability and preciseness. Furthermore, we note that cooperative and competitive influences between different transcription factor molecules of the same species are not excluded by this assumption, and we will show how to account for such effects in the individual regulation functions in the following subsections.

Linear models in which the individual regulation functions are each described by a single parameter, $r_{ij}(x_j(t)) = a_{ij}x_j(t)$, are widely used for network inference (see, for example, Chen & Church, 1999; Cohens et al., 2006; Guthke et al., 2005; Kloster et al., 2005; Sabatti & James, 2006; Vallabhajosyula et al., 2006; van Someren et al., 2006). Such linear models might be appropriate if the network under consideration operates at a specific working point, such that the system can be interpreted as the linearization around this point (Gustafsson et al., 2005; Sanguinetti et al., 2006). However, gene regulation is known to be highly nonlinear, and simple linear models are often not appropriate to capture the qualitative dynamic behavior of a system. For example, linear models have a single steady state which is either globally stable or unstable, and they cannot show complex dynamic behavior such as multi-stationarity, hysteresis or sustained oscillations.

In the next section, we will use chemical reaction kinetics to derive a more realistic parameterization of the individual regulation functions. The resulting model class is generally able to capture the mentioned, more complex behaviors.

Chemical Reaction Kinetics and the Quasi-Steady State Approximation

Following the theory of Michaelis and Menten (Michaelis & Menten, 1913; see also Alon, 2006; Yagil & Yagil, 1971), we describe binding of a transcription factor TF to a specific DNA binding site BS as a reversible chemical reaction:

$$\mathrm{TF} + \mathrm{BS} \rightleftarrows \mathrm{C} \tag{7}$$

TF and BS form a complex C with a reaction rate k_1, and this complex dissociates with a rate k_{-1}. This reaction reaches a steady state within milliseconds (Alon, 2006). Thus, the time scale for this reaction is much faster than that of the gene regulatory network, which is the scale of protein concentration changes (minutes to hours). Hence it is convenient to apply a *quasi-steady state approximation* (QSSA). In this setting, we consider slow and fast reactions on separate time scales Δt and $\varepsilon\Delta t$, $\varepsilon << 1$, respectively (Strogatz, 2000). A large difference between these scales allows for the following approximations: Con-

sidering the system on the *fast* time scale, changes of variables taking place on the slow time scale can be neglected. This means in our example, that the concentration of transcription factors in reaction (7) is treated as a constant, and the reaction approaches the *chemical equilibrium*, in which the number of complex formations equals on average the number of dissociation reactions, and the net reaction is zero. Thus, in chemical equilibrium, the ratio of reactant and product concentrations is constant. According to the law of mass action, this ratio is determined by

$$\mathrm{K} = \frac{k_1}{k_{-1}} = \frac{[\mathrm{C}]_s}{[\mathrm{TF}]_s[\mathrm{BS}]_s}. \tag{8}$$

Here, $[\mathrm{X}]_s$ is the equilibrium concentration of component X, and K is called *equilibrium constant*. It is a measure for the affinity of a DNA binding site to a transcription factor.

On the *slow* time scale in turn, the fast reaction is assumed to be always in a steady state, which, since it depends on the concentration of the transcription factors, changes slowly. It is for this reason that this approximation is called quasi-steady state approximation.

The QSSA is generally the basis for the inference of ODE model parameters from time series concentration data. It is required for a functional relation between the rate of change of the system's state at time *t* and its current state at time *t*, which is postulated in each ODE model.

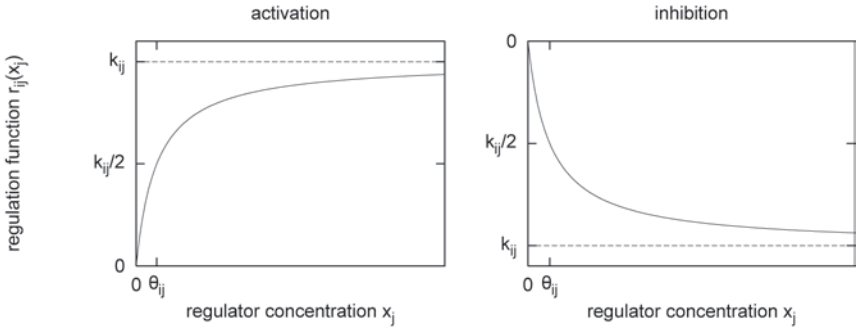
Assuming that the number of transcription factors bound to the DNA is much smaller than the number of unbound ones allows it to write the fraction of occupied binding sites among all sites, $[\mathrm{BS}]_b/[\mathrm{BS}]_t$, as a function of the total transcription factor concentration $[\mathrm{TF}]_t$ and the equilibrium constant K:

$$\frac{[\mathrm{BS}]_b}{[\mathrm{BS}]_t} = \frac{[\mathrm{TF}]_t}{[\mathrm{TF}]_t + \mathrm{K}^{-1}} \tag{9}$$

We assume this fraction to be proportional to the effect of the transcription factor on the expression rate of the regulated gene, leading to a hyperbolic individual regulation function

$$r_{ij}(x_j) = k_{ij}\frac{x_j}{x_j + \theta_{ij}} \tag{10}$$

Figure 1. Activating (left) and repressing (right) individual regulation functions, showing the effect of the regulated gene versus the regulator concentration



shown in Figure 1. The *regulation strength* k_{ij} is the maximal effect of regulator *j* on variable x_i, which is approximated for high regulator concentrations, when the fraction of binding sites bound by a transcription factor is nearly 1. The regulation strength is positive if *j* activates *i*, it is negative if *j* is an inhibitor of *i*, and zero if *j* does not have an influence on *i*. The parameter θ_{ij} is related to the equilibrium constant K and serves as a *threshold value*. If the regulator concentration x_j equals θ_{ij}, the effect on the regulated gene is half of the regulation strength k_{ij}.

Interactions Between Transcription Factor Molecules

So far, we assumed independent binding of all transcription factor molecules to their respective DNA binding sites. A more realistic description accounts for influences among different transcription factor molecules of the same species. Many transcription factors only become active as complexes, often they form dimers consisting of two molecules, or tetramers, which contain four molecules of the same species (Alon, 2006; Lu et al., 2006; Savageau & Alves, 2006). The corresponding chemical reaction then reads

$$\mathrm{mTF} + \mathrm{BS} \rightleftarrows \mathrm{C}, \tag{11}$$

where m is the number of molecules in the complex. Applying the same transformations as above, we arrive at sigmoid individual regulation functions

$$r_{ij}(x_j) = k_{ij} \frac{{x_j}^{m_{ij}}}{{x_j}^{m_{ij}} + {\theta_{ij}}^{m_{ij}}}, \tag{12}$$

which differ from equation (10) by the *Hill coefficients* m_{ij}.

The parameters k_{ij}, θ_{ij}, and m_{ij} in equation (12) can sometimes be determined empirically, and the m_{ij} may be fractional numbers. They then account for influences among transcription factor molecules in a more general way. Binding of a single molecule can, for example, facilitate binding of a second molecule, expressed by a Hill coefficient $m_{ij} > 1$. It can also have the opposite effect, and $0 < m_{ij} < 1$ in this case.

Figure 2. Hill regulation functions according to equation (12) for different Hill coefficients. The plot shows how the Hill coefficient determines the steepness of the sigmoid.

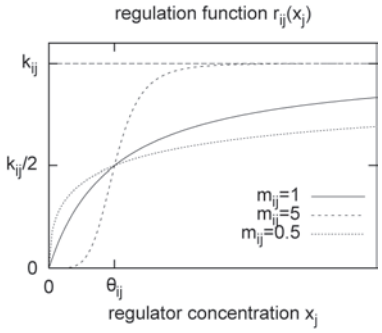


Figure 2 shows regulation functions with fixed θ and *k* and different Hill coefficients. The coefficient m_{ij}=1 corresponds to the hyperbolically increasing function in Figure 1. Increasing m_{ij} causes a sigmoid shape. Here, the role of θ_{ij} as a threshold value becomes evident: Comparing two regulation functions $r_1(x)$ and $r_2(x)$ with Hill coefficients m_1 and $m_2 > m_1$, function $r_2(x)$ is below $r_1(x)$ for $x<\theta$, and it exceeds $r_1(x)$ for $x>\theta$. Moreover, in the limit $m\to\infty$, $r(x)$ approaches a step function, which is constantly zero for $x<\theta$, and *k* for $x>\theta$, leading to piecewise constant regulation functions. Such a description has been used to model gene regulatory networks by Mestl et al. (1995) and de Jong and coworkers (de Jong et al., 2003; de Jong et al., 2004; de Jong et al., 2000).

Properties of the Model

We consider the additive ODE model

$$\dot{x}_i(t) = s_i - g_i x_i(t) + \sum_{j=1}^{n} r_{ij}(x_j(t)), \quad i = 1,\ldots,n, \tag{13}$$

with individual regulation functions given by equation (12), and point out two properties which distinguish this model from simple linear models.

First, trajectories of system (13) monotonically approach a trapping region bounded by lower and upper values

$$x_i^{\min} = \tfrac{1}{\gamma_i}(s_i + \sum\nolimits_{-k_{ij}\in\mathbb{R}_+} k_{ij}) \quad \text{and} \quad x_i^{\min} = \tfrac{1}{\gamma_i}(s_i + \sum\nolimits_{k_{ij}\in\mathbb{R}_+} k_{ij}). \tag{14}$$

This is a very pleasant property from a biological and a mathematical point of view. All concentrations are bounded for arbitrary initial conditions. This is biologically plausible. Moreover, the long term behavior is completely determined by limit sets in the trapping region, which can simplify the analysis considerably.

Secondly, the model is able to capture complex dynamic behavior such as the existence of multiple stable steady states and sustained oscillations, which are known to be related to circuits in the interaction graph (Gouze, 1998; Thieffry, 2007; Thomas, 1998; Thomas & D'Ari, 1990; Thomas et al., 1995). For example, a positive circuit is a necessary condition for the existence of multiple steady states, which are related to hysteresis, bi-stability and switch like behavior. Negative feedback is in turn required for stable periodic behavior.

BAYES REGULARIZED ORDINARY DIFFERENTIAL EQUATIONS

We will now show how to embed such an ODE model into a stochastic framework. This approach allows it to keep the quantitative accuracy of differential equation models (with underlying biochemical reaction kinetic), and at the same time account for noise in the experimental data. Furthermore, using Bayes' theorem, prior information on the biological network can be included in the inference process through a

prior distribution on model parameters. This provides a very effective way to deal with underdetermined optimization problems and overfitting so often encountered with any quantitative kinetic model.

Integration into a Probabilistic Framework

The key assumption we make in order to integrate the system of differential equations (13) into a probabilistic context is, that the ODE system (13) describes the true state of the genes x at any given time point, but that we can observe only a corrupted version

$$y(t) = x(t) + \xi. \tag{15}$$

Here, x is the vector of true concentrations of the genes, y is the vector of observations, and ξ is a vector of mean-zero, normally distributed random variables capturing noise. For simplicity, we assume the same variance σ^2 for all genes. The assumption of normally distributed noise is justified if we assume the noise to stem from many independent sources, and clearly other models are feasible at this point. We note here again that this model does not account for noise due to biological variation, since ξ is not fed back into the differential equations and does not affect the true state $x(t + \Delta t)$ of the system at a later time point $t+\Delta t$.

Given parameters $w = (s, \gamma, k, m, \theta)$ of the differential equation model, the unknown true state $x(t)=\Phi(x_0,t)$ is uniquely determined by the time t and the state x_0 at an initial time point t_0. In order to approximate this function Φ for a state $x(t+\Delta t)$ and $x_0 = x(t)$, we have to integrate equation (13) numerically. This can be done, for example, by a simple Euler discretization with fixed step size Δt,

$$x_i(t + \Delta t) = x_i(t) + \Delta t \cdot f_i(x(t)). \tag{16}$$

The probability of observing gene i in state $y_i(t + \Delta t)$ at time $t + \Delta t$ is then given by

$$p(y_i(t + \Delta t) \mid x(t), w, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\tfrac{1}{2\sigma^2}\left(h_i(x(t), w, \Delta t) - y_i(t + \Delta t)\right)^2\right], \tag{17}$$

with $h_i(x(t),w,\Delta t) = x_i(t+\Delta t)$ given by equation (16).

The conditional probability distribution (17) describes a normal distribution centered at the true state $x(t + \Delta t)$, which we approximate by the previous state $x(t)$ and an Euler step.

The conditional probability distribution of observing $y(t + \Delta t)=(y_1(t + \Delta t), \ldots, y_n(t + \Delta t))$ at time $t + \Delta t$, given true states $x(t)=(x_1(t), \ldots, x_n(t))$, is then given by

$$p(y(t + \Delta t) \mid x(t), w) = \prod_{i=1}^{n} p(y_i(t + \Delta t) \mid x(t), w). \tag{18}$$

Finally, the probability of observing a time series $Y = \{y(t_1), y(t_2), \ldots, y(t_T)\}$ of measurements at T distinct time points spaced evenly at intervals Δt, given the model parameters w and the true model states $X = \{x(t_1), x(t_2), \ldots, x(t_T)\}$ is given by

$$p(Y \mid w, X) = p(y(t_1) \mid x(t_1)) \prod_{\tau=2}^{T} p(y(t_\tau) \mid x(t_{\tau-1}), w), \tag{19}$$

where $p(y(t_\tau)|x(t_\tau))$ is a normal distribution with variance σ^2. As a function of the parameters *w* and for fixed dataset Y, equation (19) is called *likelihood*, since it describes how likely it is to observe the data *Y* if the system's true states are X.

The probability distribution (19) depends on the true states *X*, which we approximate by the empirical estimates $\hat{x}(t) = y(t)$. Using this approximation, the resulting model is equivalent to a dynamic Bayesian network, as we will detail in the end of this chapter. A computationally more expensive approach would compute $x(t_\tau)$ from $x(t_0)$ using the ODE model, and estimate $x(t_0)$ from the full set of observations for all time points. For notational convenience, we will write the likelihood as $p(Y \mid w)$ in the following, neglecting details of the implied estimation of *X* from *Y*.

Maximum Likelihood Parameter Estimation

A *maximum likelihood* approach to parameter estimation would now maximize $p(Y \mid w)$ with respect to the model parameters *w*, that is, find the model parameters *w* which maximize the probability of seeing the data. The computation is much simplified by taking the negative logarithm of the objective function (19), thus minimizing

$$l_Y(w) := -\ln(p(Y \mid w)) = -\ln\Big(p(y(t_1) \mid x(t_1))\Big) + \sum_{\tau=2}^{T}\sum_{i=1}^{n}\Big(-\ln(p(y_i(t_\tau) \mid x(t_{\tau-1}), w, \sigma)\Big). \tag{20}$$

Since our interest lies in minimizing $l_Y(w)$ with respect to *w*, we can neglect terms independent of *w*. We furthermore fix the noise level σ to 1 in the following. This parameter does not change the location of the minimum $\hat{w}_{MLE} = \arg\min_w l_Y(w)$. It will however become a relevant parameter later in the Bayesian framework.

Substituting from (17), we can simplify the last term in (20) further, and dropping terms independent of *w*, the optimization problem becomes

$$\hat{w}_{MLE} = \arg\min_w \sum_{\tau=2}^{T}\sum_{i=1}^{n} \tfrac{1}{2}\Big[h_i(x(t_{\tau-1}), w, \Delta t) - y_i(t_\tau)\Big]^2. \tag{21}$$

Diverse algorithms can be used to carry out this optimization, for example genetic algorithms (Rechenberg, 1973), simulated annealing (Kirkpatrick et al., 1983) or procedures based on gradient descent. We use the latter, as described in Press et al. (2002).

Bayesian Learning Framework

Although we have derived equation (21) from a statistical perspective, the resulting optimization problem turns out to be equivalent to classical minimum-squared-error fitting of a model to experimental data. This is an interesting result from a theoretical point of view, since it provides a connection between least squares fitting and maximum likelihood estimation for normally distributed error terms ξ. In contrast to least squares fitting, however, the statistical approach provides a straightforward framework to include

additional knowledge in the network inference process, as we will show in the following. Such additional knowledge will become of highest importance in particular in the typical setting of larger networks and only insufficient amounts of experimental data, resulting in underdetermined optimization problems in model fitting.

The main tool to address this point is Bayes' theorem, which states that, given experimental measurements *Y*, the probability distribution over model parameters *w* is given by

$$p(w \mid Y) = \frac{p(Y \mid w)p(w)}{p(Y)}. \tag{22}$$

Here, $p(Y|w)$ is the likelihood (19), $p(w)$ is a *prior distribution* over the model parameters *w*, and $p(Y) = \int_{\Omega} p(Y \mid w)p(w)dw$ is a normalizing factor called *evidence*. $P(w|Y)$ is called *posterior distribution*. It describes the probability distribution of the model parameters *w*, given the experimental observations *Y*.

The prior distribution $p(w)$ over the model parameters *w* can be used to integrate additional biological knowledge into the learning process. Imoto et al. (2003), in their pioneering work, demonstrate this by expressing the prior knowledge over interactions between specific genes in terms of energy functions, from which a prior distribution over network structures is obtained in the form of a Gibbs distribution.

We will assume less explicit prior knowledge in the following. Instead of considering explicit knowledge of the form "there should be an edge between gene A and B with high probability", we will only define a vague prior of the form "the network should be sparse, that is, it contains only few edges relative to the fully connected network". This is biologically motivated in so far as it is highly unlikely that there are direct regulatory interactions between most pairs of genes in the network.

In terms of the differential equation model (13), this sparseness assumption translates into the assumption that most of the parameters k_{ij} should be equal to or almost equal to zero. We therefore use a mean-zero normal distribution with variance σ_{ij}^2 as prior distribution on the k_{ij},

$$p(k_{ij} \mid \sigma_{ij}) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left[-\frac{k_{ij}^2}{2\sigma_{ij}^2}\right] \tag{23}$$

This will assure that the k_{ij} do not become arbitrarily large, however, it does not yet enforce sparseness in the sense that most of these parameters should be close to zero. To enforce the latter, we specify a second level of prior distribution over the standard deviations σ_{ij}. We would like most of the normal distributions to be strongly concentrated around their mean zero; hence their standard deviation should be small. This is expressed using a *gamma distribution*,

$$p(\sigma_{ij} \mid a, r) = \frac{a^r \sigma_{ij}^{r-1}}{\Gamma(r)} e^{-a\sigma_{ij}}, \tag{24}$$

where $\Gamma(r) = \int_0^{\infty} t^{r-1}e^{-t}dt$ is the gamma function and $1/a$ and *r* are scale and shape parameters.

We can now compute the prior distribution $p(k_{ij}|a,r)$ over k_{ij} by integrating out σ,

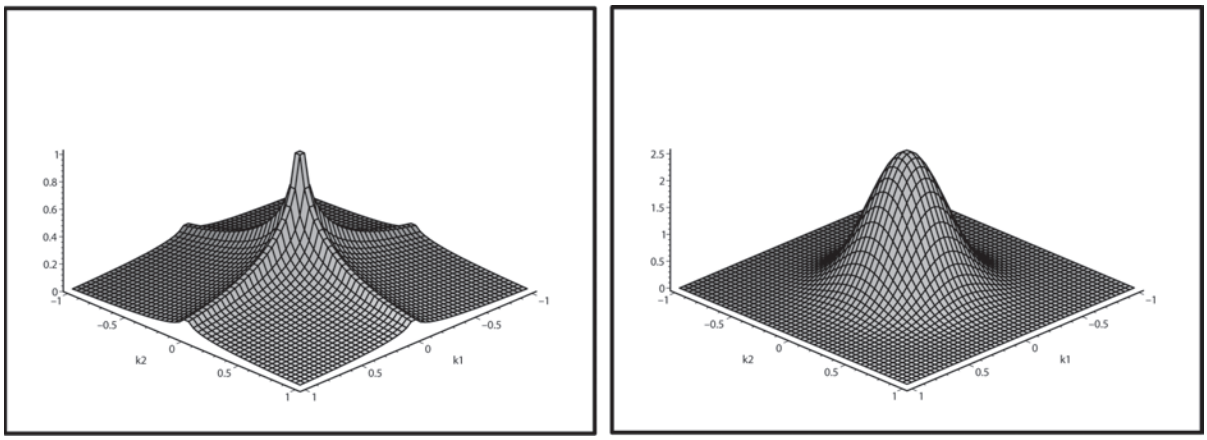
$$p(k_{ij} \mid a, r) = \int_0^\infty p(k_{ij} \mid \sigma) p(\sigma \mid a, r) d\sigma. \tag{25}$$

Although this integral is not analytically tractable, it can be approximated very well numerically using Gauss-Laguerre quadrature. This prior distribution is shown in the two-dimensional case in Figure 3 (right). In comparison to the normal distribution (Figure 3 left), it can be clearly seen how this prior favors sparse solutions in the sense that only few of the k_{ij} are significantly distinct from zero. We note in passing that a very similar effect can be obtained using a prior based on the L_q-norm for $q<1$. Also, this prior enforces much stronger sparseness constraints than the Laplace prior traditionally used for this purpose.

For the synthesis and degradation rates s_i and γ_i, we will use independent gamma distributions $p(s_i|a_{si},r_{si})$ and $p(\gamma_i|a_{\gamma i},r_{\gamma i})$ as prior. This choice is motivated from the requirement that these parameters must be positive, and they should not become arbitrarily large. For the sake of simplicity, we assume fixed values for the Hill coefficients m_{ij} and the threshold parameters θ_{ij}. These latter parameters can only be estimated well from data if sufficient time points are available, and in particular estimation of the Hill coefficients is numerically very unstable.

We are now ready to optimize the posterior $p(w|Y)$. In the following, we will show results stemming from simple maximization of $p(w|Y)$ with respect to w using conjugate gradient descent, and the respective *maximum a posteriori* (MAP) estimator is denoted $\hat{w}_{MAP}$. This has the advantage that it is relatively straightforward and easily computed, but it may suffer from problems with (multiple) local optima. Sampling from the posterior distribution using Markov-chain Monte Carlo methods and optimization using simulated annealing are alternatives that we are presently evaluating in our groups.

Figure 3. 2-Dimensional prior over network parameters k_{ij}. Left: Normal distribution, right: Prior according to equation (25). While the normal distribution penalizes the overall distance of the weights k from the origin, it does not enforce sparseness. The plot clearly demonstrates, how this prior assigns lower probability mass to points where both k_1 and k_2 are significantly distinct from zero than to points where only one of the two parameters deviates from zero, even if the total distance from the origin is the same.



RESULTS

Results on Simulated Data

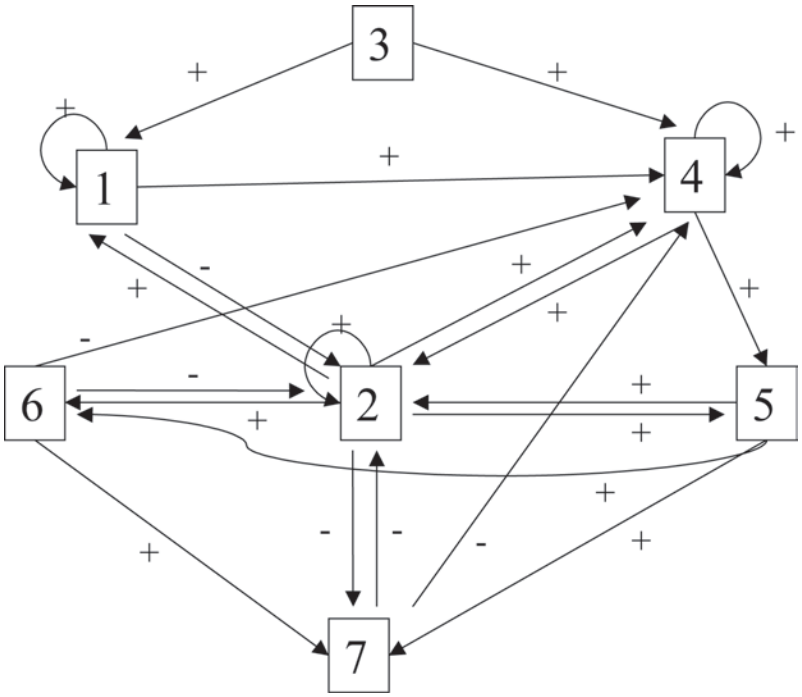
Data Simulation

We used equation (15) with the differential equation model (13) to simulate time series data for a network of seven genes. The interaction graph of the system is shown in Figure 4. Synthesis and degradation rates were set to s_i=1, γ_i=0.1 for i=1,...,7. For the parameters of the individual regulation functions we used the values θ_{ij}=5, m_{ij}=2 for i,j=1,...,7, and k_{ij}=±2 with signs according to the edge labels in Figure 4. The discretization step width was set to Δt=1. Time series data with different initial states $x_i(0)$ randomly drawn from a uniform distribution over the interval [0,5] were simulated, three time points each. We varied the noise level σ and the number of time points used to learn the model parameters.

Parameter Estimation

Conjugate gradient descent was carried out to maximize the posterior distribution $p(w|Y)$ with respect to model parameters w. We compared the maximum likelihood estimator (MLE) $\hat{w}_{MLE,}$ using different noise levels and numbers of time points for the inference process. The threshold values θ_{ij} and the Hill coefficients m_{ij} were fixed to values θ_{ij}=5 and m_{ij}=2 for $i, j=1,...,n$. To test how strongly results depend on these parameters, we compared results using several different values, and observed no significant

Figure 4. Network topology used to simulate time series data. Numbered nodes correspond to genes, edges represent regulatory interactions. Labels (+) and (-) indicate positive or negative regulation.

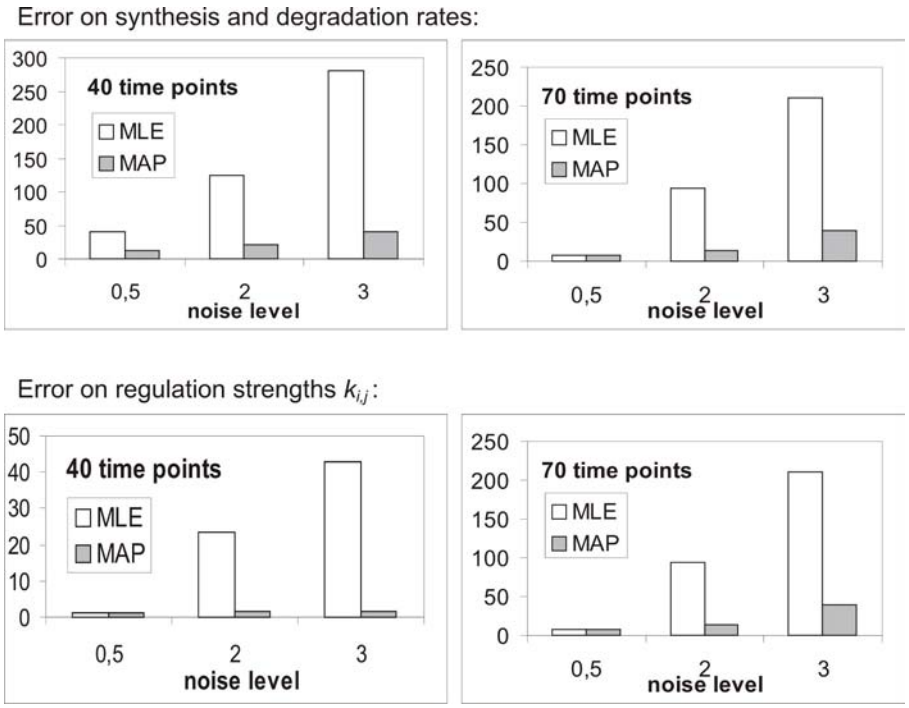


difference. Gradient descent was started with $s_i=\gamma_i=0.1$. All regulation strengths k_{ij} were initially set to 0. Parameters for the gamma distributions over synthesis and degradation rates were set to $r_{si}=2$, $a_{si}=1$, $r_{\gamma i}=1.0001$, $a_{\gamma i}=2$ for $i=1,...,n$. Parameters for the gamma distribution over standard deviations σ of the noise term ξ were set to $r=1.2$ and $a=1.5$.

Figure 5 shows mean squared errors on the estimated model parameters. The errors on synthesis and degradation rates s_i and γ_i (top) are given in percent, errors on regulation strengths k_{ij} (bottom) are given as absolute values. Shown are results for 40 time points (left) and 70 time points (right). Both approaches lead to comparable results in case the dataset is large (70 time points) and a small level of noise ($\sigma=0.5$).

Decreasing the number of time points or increasing the noise level, however, it can be observed that the Bayesian approach clearly outperforms MLE. In this setting, the maximum likelihood objective function is close to zero at $\hat{w}_{MLE}$, but the errors for estimated model parameters are huge, indicating that $\hat{w}_{MLE}$ overfits the data. The Bayesian approach regularized by the prior distribution is less prone to this problem.

Figure 5. Mean squared errors on reconstructed model parameters for simulated dataset, for maximum likelihood (MLE) and maximum a-posteriori (MAP) approaches as described in the text. The upper two plots show errors on synthesis and degradation rates, errors on regulation strengths are shown in the lower two plots. The evaluation was repeated for two different dataset sizes, 40 time points (left) versus 70 time points (right), and for three different levels of noise introduced in data simulation. Results show a clear advantage of the MAP approach in case of high noise levels and a low number of time points.



Inferred Network Structure

We performed a *receiver operator characteristics* (ROC) analysis to evaluate the topology of the inferred network. A threshold value z is used on the estimated regulation strengths k_{ij}. Component j is assumed to regulate i if the corresponding regulation strength exceeds this threshold, $|k_{ij}|>z$. ROC curves are obtained by varying z from 0 (all interaction strengths are significant, and the inferred network is fully connected) to max $\{|k_{ij}| \mid i, j=1,...,n\}$ (none of the strengths are significant, the set of edges of the inferred network is empty), and calculating sensitivity and specificity of edge recognition for the resulting networks.

Figure 6 shows ROC curves for noise levels σ=2 and 3, and 40 and 70 time points. A ROC curve of a good classifier is positioned in the upper left corner, where both specificity and sensitivity are high. Guessing edges in the network leads on average to the diagonal, where sensitivity equals 1-specificity. Analogously to the mean squared error analysis, also the ROC analysis shows that MLE fails given only 40 time points. In contrast, the MAP approach infers parts of the network structure correctly. Not surprisingly, both approaches perform better with 70 time points, but still the Bayesian approach outperforms maximum likelihood.

Figure 6. Receiver Operator Characteristics (ROC) analysis for the structure of the inferred network, for different noise levels used in data simulation and different numbers of time points available for network inference. Plotted are curves of sensitivity against specificity for the presence of edges in the network. These are computed by continuously varying the threshold on abs($k_{i,j}$) used to decide whether an edge is present or not. This analysis demonstrates the superior performance obtained using the sparseness prior (25) over a computation carried out on the likelihood alone.

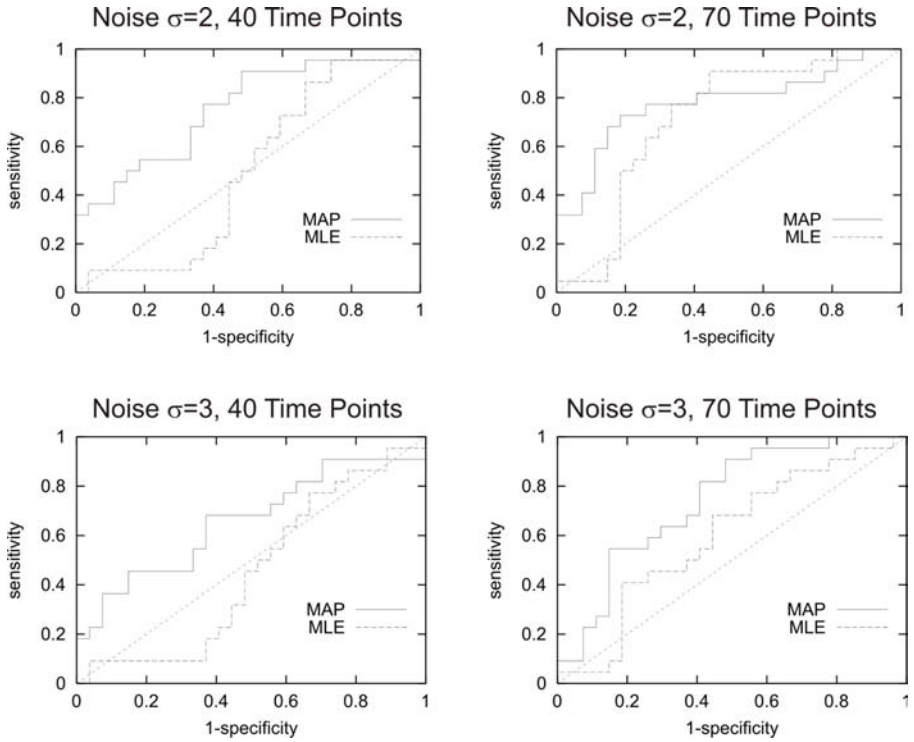
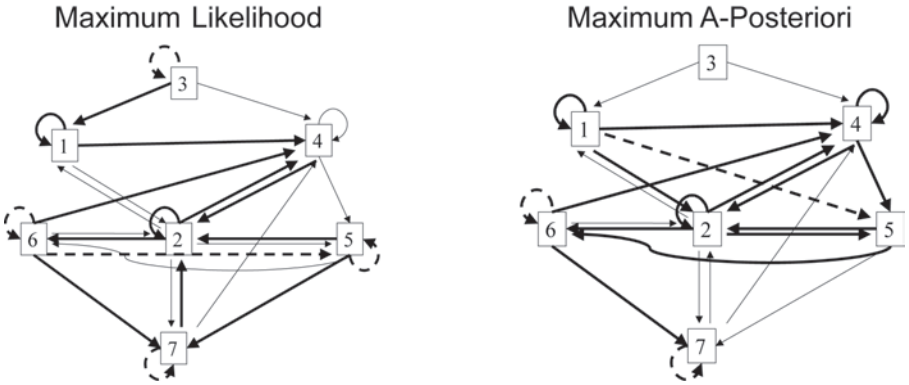


Figure 7. Inferred networks on simulated dataset with 70 time points and noise level σ=2, obtained as described in the text. The left plot shows the network inferred using the maximum likelihood approach, the right plot shows the network computed from maximum a-posteriori. Included are in each plot the 17 edges with highest weights, marked in bold. Solid bold lines indicate true positives, dashed bold lines false positives. Thin lines indicate edges were no edge was inferred, although an edge between the involved nodes was present in data simulation.



In Figure 7 we show the inferred networks for 70 time points and noise level σ=2. The 17 edges with highest weights are marked in bold. Solid lines indicate true positives, dashed lines false positives, and thin lines false negatives. 12 of these 17 edges are true positives in the MLE network (Figure 7 left), and 14 true edges are found in the Bayesian approach (Figure 7 right).

The *area under the curve* (AUC) value is a measure for the overall performance of the classifier, independent of the threshold value *z*. It is computed from ROC curves by integrating over the curve to calculate the area under the curve. The AUC is a value between 0 and 1, it increases with increasing performance. AUC values for both estimators $\hat{w}_{MLE}$ and $\hat{w}_{MAP}$ can be seen in Figure 8. The left plot shows how performance behaves when the noise level is increased, for a fixed dataset size of 70 time points. The AUC values for an increasing number of time points used for learning and a fixed noise level σ =2 are shown in the plot on the right hand side. This analysis provides information about the maximal level of noise and the minimal number of time points required for the maximum likelihood estimator to succeed. For most of the datasets considered here, the AUC value of the maximum likelihood estimator is around 0.5, and thus not better than guessing. It reaches a value of approximately 0.7 only at a noise level σ =2 and with 70 time points. Figure 8 demonstrates that the minimal number of time points needed to draw meaningful conclusions can be reduced by the Bayesian approach. Using 70 time points (left), the AUC value of $\hat{w}_{MAP}$ exceeds 0.7 for the noise levels σ =2 and 3. Increasing the noise further to σ=4, it also drops. Using the smallest noise level (right), the MAP approach is able to infer at least parts of the network structure correctly, even with only 20 time points.

A Regulatory Network of the Yeast Cell Cycle

We applied the approach presented to the microarray study of the *Saccharomyces cerevisiae* cell cycle by Spellman et al. (1998). This dataset consists of four gene expression time series from four different synchronization protocols and contains 69 time points in total, collected over eight cell cycles. We in-

Figure 8. Area under the ROC curve (AUC) values for maximum likelihood and maximum a-posteriori network structures inferred, obtained by integrating the area under the ROC curves. The figure shows AUC values for different noise levels with fixed number of time points (=70) in the left plot, and different dataset sizes (number of time points) used in network inference for a fixed noise level σ=2.

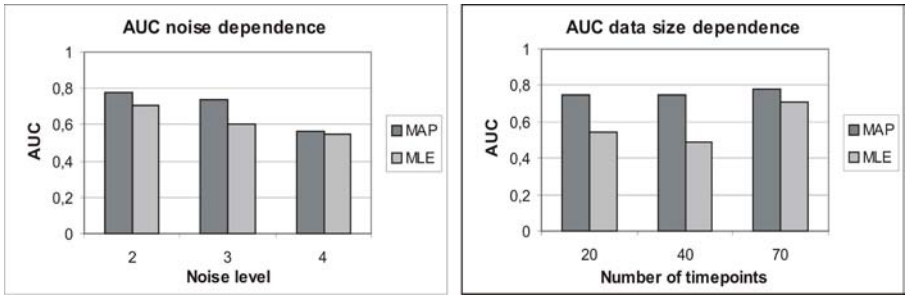
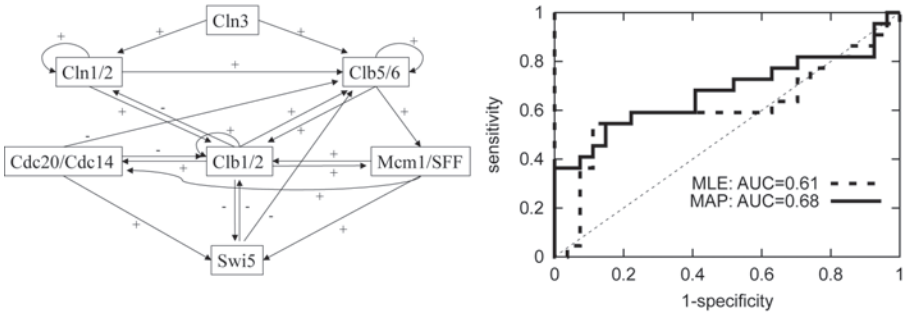


Figure 9. Left: Network compiled from the literature which was used to evaluate the inferred network topologies. Right: ROC curves for inferred network topologies, obtained using the maximum likelihood (dotted line) and the maximum a-posteriori (solid line) approach.

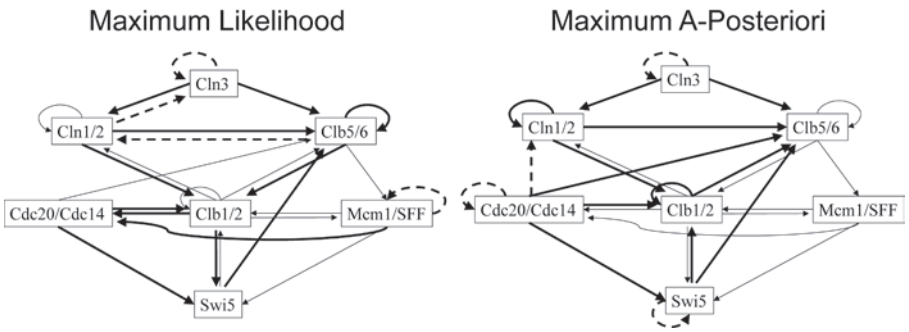


cluded eleven genes into our analysis, which are known to be involved in the yeast cell cycle (Li et al., 2004). The reference network in Figure 9 (left) was used for evaluation. It is a reduction of the regulatory network specified in Li et al. (2004). Details on this network are given in Radde & Kaderali (2007). The eleven genes are combined into seven nodes in the network. Time series data of a node which contains more than one gene is represented by means of measurements. Missing values were replaced by means of concentrations of consecutive and subsequent time points.

The threshold values θ_{ij} and the Hill coefficients m_{ij} were fixed to values $\theta_{ij}=1$ and $m_{ij}=2$ for $i,j=1,...,n$. Gradient descent was started with $s_i=\gamma_i=0.1$. All regulation strengths k_{ij} were initially set to 0. Parameters for gamma distributions over synthesis and degradation rates were set to $r_{si}=0.01$, $a_{si}=0.1$, $r_{\gamma i}=0.01$, $a_{\gamma i}=0.1$ for $i=1,...,n$. Parameters for the gamma distribution over standard deviations σ were set to $r=1.7$ and $a=5$.

Figure 9 (right) shows ROC curves for the ML and the MAP approach with AUC values 0.61 and 0.68, respectively. Some of the main regulatory interactions are revealed in both approaches, but here as well the Bayesian approach clearly outperforms maximum likelihood. Inferred network structures are presented in Figure 10 for $\hat{w}_{MLE}$ (left) and $\hat{w}_{MAP}$ (right). The 16 edges with highest estimated inter-

Figure 10. Inferred network structures, as derived using the maximum likelihood approach (left) and the maximum-a-posteriori approach (right). Bold solid lines indicate true positives, dashed bold lines are false positives. Thin lines correspond to regulatory interactions reported in the literature, which were not learned in network inference.



action strengths are shown in bold. True positives are drawn as continuous bold lines, and dashed bold lines correspond to false positives. Thin lines are interactions which were described in the literature, but were not revealed in our approach. In both networks, 12 of 16 regulations found are true positives. The Bayesian estimator $\hat{w}_{MAP}$ reveals more regulations between different genes than $\hat{w}_{MLE}$, but it also reports a couple of artificial self-regulations. Seven of these 12 true positives appear in both networks. These include regulations that involve main cell cycle transcription factors, for example:

- Cln3 → Cln1/2 and Cln3 → Clb5/6: Cln3 is a key regulator of the transcription factor complex SBF, which activates expression of the genes Cln1 and Cln2, and of MBF, which activates the genes Clb5 and Clb6. Cln3 triggers entry of the cell cycle into the S- and the M-phase, respectively, by activating these complexes.
- Swi5 → Clb5/6: Swi5 is the transcription factor of Sic1, which inhibits Clb5/6.

Most of the false negative edges are indirect edges which involve the protein Sic1, a global inhibitor of several cell cycle regulated genes.

Relation to Other Stochastic Approaches

This subsection details the relation between Bayes regularized ODE models and other stochastic approaches widely used for network inference from experimental data, especially *dynamic Bayesian networks* (DBN) and models including intrinsic noise and measurement noise. Since DBNs are an extension of Bayesian networks, we start by defining a Bayesian network.

A *Bayesian network* is a stochastic model with a set $V=\{y_1,...,y_n\}$ of n random variables. The state space Ω can contain discrete and continuous y_i. Bayesian networks are static models, they do not consider time. The central assumption made in Bayesian network models is, that the *joint probability distribution* $p(\Omega)$ can be rewritten as the product of *local conditional probability distributions* $p(y_i|\text{parents}(y_i))$. The set $\text{parents}(y_i)\subseteq V\backslash\{y_i\}$ is called the *parent set* of y_i. Hence, Bayesian networks assume conditional *independence assertions* between the variables *V,* which allow it to construct the joint probability distribution over the set of variables *V* from the local ones:

$$p(y_1,\ldots,y_n) = \prod_{i=1}^{n} p(y_i \mid parents(y_i)) \tag{26}$$

The independence assertions state that a variable y_j which is a successor of y_i cannot be a predecessor of y_i at the same time. This requirement is needed to assure that equation (26) is well-defined. As a consequence, a Bayesian network contains variables with empty parent sets. The local probability distributions of these variables are said to be *unconditional*.

A Bayesian network can graphically be represented by a *directed acyclic graph* (*DAG*) $G(V,E)$, with nodes V and edges E. Nodes in this graph correspond to variables, and the set of edges indicates parent relations, $E=\{e_{ij} | \, y_j \in \text{parents}(y_i)\}$. In terms of this graph, the joint probability distribution (26) is only well-defined for acyclic graphs.

Murphy & Mian (1999) were among the first who modeled gene interactions by Bayesian networks, and they are still commonly used models today (see e.g. Bulashevska & Eils, 2005; Friedman et al., 2000; Hartemink et al., 2002; Pe'er et al., 2001). Learning a Bayesian network from data corresponds to estimating the joint probability distribution $p(Y)$, which defines the structure of the DAG. Similar to a correlation analysis, which does not provide directions of edges, the inference of a unique DAG is not always possible. Networks with the same undirected graph structure but different directions of some edges may represent the same distribution (Dojer et al., 2006). These graphs imply the same conditional independence relations, that is, they are contained in the same equivalence class, and the data do not allow for a distinction (Pe'er, 2005, Dojer et al., 2006).

The static nature and the requirement that the graph is acyclic are two main drawbacks of Bayesian network models (Husmeier, 2003). Dynamic Bayesian networks (DBNs) have been suggested to overcome both limitations (Dojer et al., 2006; Friedman et al., 1998; Husmeier, 2003; Pe'er, 2005; Zou & Conzen, 2005). In DBNs, a separate random variable $y_i(t)$ is introduced for each time point t=1,...,T, the system is thus unrolled over time. The joint probability distribution is defined over the set $y(1) \cup y(2) \cup \ldots \cup y(\mathrm{T})$ of time-dependent variables. Since such a distribution can be very complex, two simplifying assumptions are usually made in practice (Friedman et al., 1998). First, the process is assumed to be *Markovian*, that is, the probability distribution $p(y(t))$ depends solely on the previous state $y(t\text{-}1)$,

$$p(y(t) \mid y(0),\ldots,y(t-1)) = p(y(t) \mid y(t-1)). \tag{27}$$

Second, the process is *time-homogeneous*, which means that the transition probabilities $p(y(t)|y(t\text{-}1))$ do not explicitly depend on t.

Inserting the empirical estimate $\hat{x}(t_{\tau-1}) = y(t_{\tau-1})$ for the true state $x(t_{\tau\text{-}1})$ in equation (19), this probability distribution equals equation (27). In particular, it is completely determined by the observations Y and thus does not depend any more on the true states X. Therefore, using these estimates for $x(t)$, our approach is equivalent to a DBN, which does not distinguish between true states X and observations Y. However, we point out two important conceptual differences between those two models.

First, in our differential equation approach, we assume that the dynamic behavior of the network can be described deterministically, and noise in the data Y is completely due to the measurement process (compare also Golightly & Wilkinson, 2008). Hence, the noise term does not depend on the time interval between two measurements. We therefore model the noise using a mean-zero normal distribution with variance σ^2, independent of this time interval. In contrast, stochasticity in a dynamic Bayesian network is assumed to stem from intrinsic noise, that is, the system under consideration is a stochastic system.

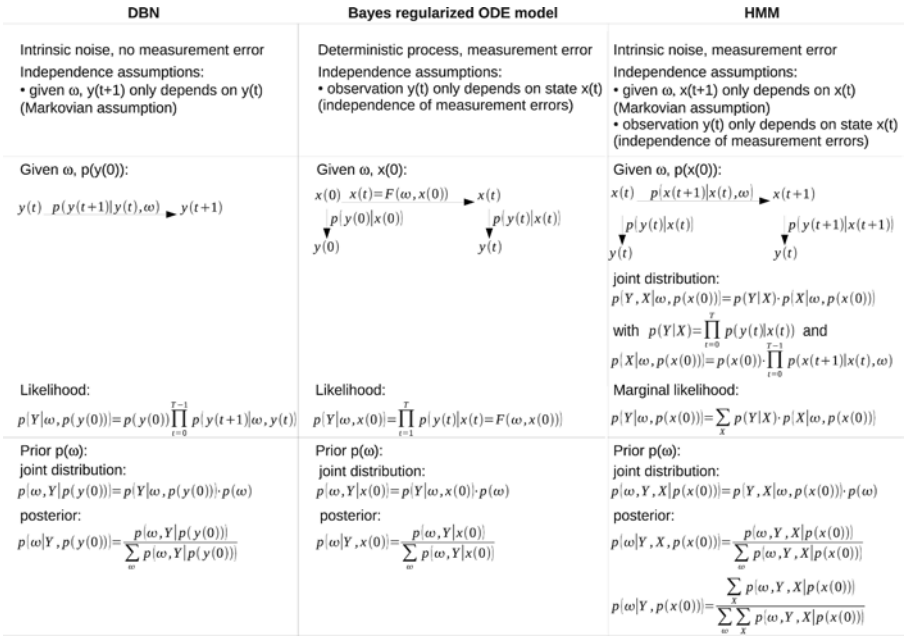
The noise level will depend on the length of the time interval between two measurements here. Consequently, a fixed noise level σ^2 as is assumed here would be justified only for equidistant time intervals in a DBN approach. This aspect is also discussed in de Hoon et al. (2003).

Second, since $x(t)$ is assumed to be determined by the time t and an initial state x_0 at an arbitrary time point t_0, according to the differential equations, any time point could be used to estimate the true state $x(t)$ at time t. This need not necessarily be done using the previous time point $x(t\text{-}\Delta t)$, which corresponds to the Markovian assumption underlying Bayesian networks. A computationally more expensive approach, for example, would compute $x(t)$ for all t as a function of $x(t_0)$ using numerical integration of the differential equations. The initial value $x(t_0)$ can then be estimated from the data simultaneously. For a related discussion we refer to Peifer and Timmer (2007).

Figure 11 illustrates the concepts underlying DBNs, the ODE approach introduced here, and a modeling approach, for example a Hidden Markov model, including stochasticity due to both biological variation and measurement errors. Shown are the respective independence assumptions (first row), the relation of model variables via functions or probability distributions and the likelihood for given parameters and initial conditions (second row), and the posterior distributions in a Bayesian framework when including a prior distribution $p(w)$ on model parameters (third row).

In DBNs, the state $y(t)$ of the system, which is described as a random variable depending on the previous state $y(t\text{-}\Delta t)$, can directly be observed. In contrast, our ODE approach distinguishes between the state $x(t)$ of the system, which is deterministically determined by the systems of differential equations, and the

Figure 11. Schematic of differences between dynamic Bayesian networks (DBN), Bayes regularized ODE models as used in this work, and a Hidden Markov model (HMM) including intrinsic noise and measurement noise. The first column lists the different independence assumptions, which are graphically illustrated together with the likelihoods in column two. The posterior distributions in a Bayesian framework are given in the third column.



observation $y(t)$, which is corrupted by measurement noise. Hence $y(t)$ is a random variable depending on the true state $x(t)$. Using numerical integration to express $x(t)$ as a function of the previous state $x(t\text{-}\Delta t)$ and inserting the empirical estimate $\hat{x}(t-\Delta t) = y(t-\Delta t)$, the model is equivalent to a DBN. Finally, the figure also shows a model which captures intrinsic noise stemming from biological variation and noise due to the measurement process. As in the DBN, the true state $x(t)$ of the system corresponds to a random variable with conditional distribution $p(x(t)|x(t\text{-}\Delta t))$. Similar to our approach, observations $y(t)$ are also random variables with distributions depending on these true states $x(t)$. Maximizing the probability p(Y) in this model requires averaging over the unknown states X. This p(Y) is called *marginal likelihood* and usually requires sophisticated sampling methods.

There is currently an ongoing discussion whether regulation of gene expression should be described deterministically or stochastically (Srivastava et al., 2002). Supporter of the stochastic side argue that the regulation of gene expression via binding of a transcription factor to the DNA is a random discrete process, and likewise the production of proteins. The difference becomes particularly evident if the molecules involved in the reactions are present in low copy numbers, resulting in a large phenotypic variability among different cells of the same population (Elowitz et al., 2002; Ozbudak et al., 2002; Raser & O'Shea, 2004). A prominent example of a bistable outcome that is driven by noise is the switch between lytic and losygenic states in bacteriophage-λ (Raser & O'Shea, 2004; Ozbudak et al., 2002). Here, a positive auto-regulation of the gene cI is assumed to amplify the effect of initially small variations. Phenomena such as the loss of synchrony of circadian clocks and a decrease in the precision of cell signals are also ascribed to the influence of noise (Ozbudak et al., 2002).

Several probabilistic modeling approaches have been developed to account for stochasticity in gene expression (Blake et al., 2003; Chen et al., 2005; Goutsias & Kim, 2006; Raser & O'Shea, 2004). Here, binding of a transcription factor to a promoter is modeled as a discrete process, and the reaction rates are related to the probabilities for complex formation and dissociation. These models describe the behavior of single cells, and results can, for example, be used to study the heterogeneity in cell populations.

Supporters of deterministic ODE models argue that these models correspond to the average behavior of a large number of cells, such that concentration changes can continuously be described. Furthermore, they presume a deterministic overall behavior of a cell population. This is of course a simplification, but many high-throughput data do not provide information about the level of noise in cells, and hence parameters of stochastic models cannot be estimated from these data. In this sense, using continuous and deterministic models to infer regulatory networks from microarray data can be seen as a data-driven approach, which reflects the level of information included in the data.

Concluding, it is an interesting issue to understand the role of stochasticity in connection with regulatory network structures. On the one hand, many biological networks are believed to be optimally designed for a reliable and robust functioning under considerable intrinsic and extrinsic noise, which means that they are able to suppress noise, supporting the deterministic approaches. On the other hand, an amplification of noise in the cellular networks is sometimes desired for a rich diversity among individual cells (Elowitz et al., 2002; Raser & O'Shea, 2004).

CONCLUDING REMARKS

Summary

We have presented an approach for the inference of gene regulatory networks from time series data, which is particularly tailored to handle sparse datasets.

Using chemical reaction kinetics, we derived a differential equation model for gene regulatory networks, which describes the influence of a gene product on the expression rate of another gene by sigmoid functions. In order to estimate model parameters from experimental data, this model was embedded into a stochastic framework. Observations were interpreted as

realizations of random variables whose underlying distributions are determined by the differential equations model and a stochastic noise term. We defined a Bayesian framework by specifying prior distributions over network parameters, which reflect prior beliefs about the model parameters before having seen the data. We analyzed the posterior distribution over model parameters; this posterior reflects the knowledge about true parameter values after having taken the observations into account.

Our method was evaluated on simulated data first. Here, we focused on the relation between size and quality of the dataset and the respective outcome. Several datasets with varying numbers of time points and different levels of noise were analyzed. Results were compared with the classical maximum likelihood approach.

While both approaches give similar results in case of optimal datasets with a large number of time points and a low noise level, they differ considerably in the setting of sparse and noisy data. Here, the maximum likelihood approach adapts to specific random features of the dataset which are not related to the overall structure of the system. As a consequence, the variance of results obtained from different datasets is large. Furthermore, in each case the value of the optimized likelihood function is rather small, but at the same time errors of estimated model parameters are large and, correspondingly, the inferred network structures are wrong. The results of the Bayesian approach in turn show less variance due to the regularization of learning by the prior distributions over model parameters. Our analysis shows that results can be improved, yielding a higher quality of the inferred network structure, by relatively general prior distributions enforcing sparseness.

We furthermore applied our approach to a real dataset on the yeast cell cycle. We inferred some of the main interactions reported in the literature with both the maximum likelihood and the Bayesian approach, the latter approach outperforming maximum likelihood.

Although the Bayesian approach is superior to maximum likelihood estimation for the real dataset, results are not as good as for simulated data for several reasons. The most obvious difference between the analysis of simulated and real data is the relation between the real system and the model class that is used to describe underlying processes. While a specific model of this class was used to carry out simulations and create artificial datasets, the consequences of simplifications which are included in the model class are not always obvious for the real system. For example, the dataset of the *Saccharomyces cerevisiae* cell cycle contains measurements of mRNA concentrations, which do not provide information about post-transcriptional regulation processes. As already mentioned, a further simplification is the assumption of additivity of influences from different transcription factors, which may not always be justified for real world systems.

Finally, we compared our approach with related stochastic approaches currently used for network inference. The relations are summarized in Figure 11.

Discussion and Future Research Directions

To conclude, we point out some general aspects of the Bayesian approach presented and discuss further research directions.

Results of a Bayesian approach depend on the prior distribution $p(w)$, and different methods have been suggested to determine parameters of this distribution. For a review we refer to Kass & Wassermann (1996). Generally, the more $p(w)$ deviates from a uninformative flat distribution, the more influence relative to the likelihood does it have on the posterior

distribution, dominating over the information provided by the data. Using a very strong prior distribution on the one hand, a good choice of $p(w)$ can improve results significantly, but an improperly chosen prior will cause a large bias. On the other hand, the results of the Bayesian and the maximum likelihood approach are similar when using a flat prior with marginal influence on the posterior. Such a distribution might not prevent overfitting in case of small datasets, but it also causes less biased results.

Different methods have been proposed to determine prior distributions, taking different types of information into account (see for example Beal et al., 2005; Li et al., 2002; Rogers and Girolami, 2005). Prior knowledge in terms of information provided by additional data sources has, for example, been used by Bernard & Hartemink (2005); Imoto et al. (2003) and Werhli & Husmeier (2007). Empirical Bayes methods (Gelman et al., 2003, Maritz and Lwin, 1989) choose hyperparameters in dependence of the data, for example by maximizing the posterior with respect to both the model parameters and hyperparameters. Compared to our approach, which fixes $p(w)$ in advance, both the likelihood and the prior terms are influenced by the data here. A common approach to investigate the amount of information contained in the data is to examine the sensitivity of the posterior distribution with respect to changes in the prior (Lavine, 1999).

A further aspect we would like to elaborate on concerns the conclusions we draw from analyzing the posterior distribution. We searched for the maximum of this distribution, leading to a point estimate $\hat{w}_{MAP}$. However, this distribution can provide far more information. Considering the entire distribution, its variance, for example, is a measure of the reliability of the results. While a simple maximization can be done with gradient methods, such a more comprehensive analysis would require methods to sample from the unknown posterior distribution. An estimate of the whole distribution permits statistical statements such as “The expectation value of a certain parameter is 5 with a low variance, we can be relatively sure about this result” or “The evidence is small, which reflects that the information of the prior distribution and the data are contradictory.”

A critical point of our approach is the scalability to larger networks in practice. The number of putative interactions increases quadratically with the number of network components, and currently available datasets typically only allow for the reconstruction of networks with a few dozen components at most. This is, however, a problem in each quantitative dynamic modeling approach and can only be faced by larger datasets or reliable prior information about the network at hand.

Finally, we point out two general future research directions in the field of gene regulatory, or, more generally, biochemical networks. First, besides deciding whether a network component regulates another one, i.e. learning the topology of the interaction graph, it is an important issue to explain mechanisms causing the qualitative dynamic behavior of the system. Feedback mechanisms are especially interesting in this light, since they are necessary for complex behavior such as oscillations and multi-stationarity. Differential equations are particularly well suited to capture the dynamic behavior of a system, and thus our approach seems to be promising in this setting.

The second direction addresses the robustness of networks concerning their functionality under considerably varying external conditions and intrinsic variability, as already mentioned above. During evolution, most organisms seem to have built up complex mechanisms which make regulation processes within a cell robust against perturbations (Kitano, 2002). Thus, while reaction rates of single reactions can vary in a wide range, the overall response of a cellular network is often extremely stable. Thus, the networks have the ability to compensate for stochastic fluctuations, a property which highly justifies our deterministic model over a completely stochastic approach. An understanding of this robustness will be an important step towards a more comprehensive understanding of cellular regulation mechanisms (Kitano, 2007).

REFERENCES

Alon, U. (Ed.). (2006). *An introduction to systems biology–design principles of biological circuits.* Mathematical and Computational Biology Series. London, UK: Chapman & Hall/CRC.

Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C., & Wild, D. L. (2005). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics (Oxford, England), 21*(3), 349–356. doi:10.1093/bioinformatics/bti014

Bernard, A., & Hartemink, A. J. (2005). Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. *Pac. Symp. Biocomp.* (PSB05) (pp. 459-470). NJ: World Scientific.

Blake, W. J., Kaern, M., Cantor, C. R., & Collins, J. J. (2003). Noise in eukaryotic gene expression. *Nature*, *422*, 633–637. doi:10.1038/nature01546

Bulashevska, S., & Eils, R. (2005). Inferring genetic regulatory logic from expression data. *Bioinformatics (Oxford, England), 21*(11), 2706–2713. doi:10.1093/bioinformatics/bti388

Chen, K.-C., Wang, T.-Y., Tseng, H.-H., Huang, F., & Kao, C.-Y. (2005). A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae. *Bioinformatics (Oxford, England)*, *21*(12), 2883–2890. doi:10.1093/bioinformatics/bti415

Chen, T., He, H. L., & Church, G. M. (1999). Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *4*, 29–40.

Cokus, S. J., Haynor, D., Gronbech-Jensen, N., & Pellegrini, M. (2006). Modelling the network of cell cycle transcription factors in the yeast *Saccharomyces cerevisiae. BMC Bioinformatics*, *7*(381).

De Hoon, M. J. L., Ott, S., Imoto, S., & Miyano, S. (2003, September 27-30). Validation of noisy dynamical system models of gene regulation inferred from time-course gene expression data at arbitrary time intervals. *Poster Proceedings of the European Conference on Computational Biology (ECCB 2003)*, Paris, France.

De Jong, H., Geiselmann, J., Hernandez, C., & Page, M. (2003). Genetic network analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics (Oxford, England), 19*(3), 336–344. doi:10.1093/bioinformatics/btf851

De Jong, H., Gouze, J.-L., Hernandez, C., Page, M., Sari, T., & Geiselmann, J. (2004). Qualitative simulation of genetic regulatory networks using piecewise linear models. *ull. Math. Biol.*, *66*(2), 301–340. doi:10.1016/j.bulm.2003.08.010

De Jong, H., & Page, M. (2000). Qualitative simulation of large and complex genetic regulatory systems. In W. Horn (Ed.), *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI2000)* (pp. 191-195). Berlin: IOS press.

Dojer, N., Gambin, A., Mizera, A., Wilczynski, B., & Tiuryn, J. (2006). Applying dynamic Bayesian networks to perturbed gene expression data. *BMC Bioinformatics*, *7*(249).

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. John Wiley & Sons, 2nd edition.

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*, 1183–1186. doi:10.1126/science.1070919

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3/4), 601–620. doi:10.1089/106652700750050961

Friedman, N., Murphy, K., & Russell, S. (1998). Learning the structure of dynamical probabilistic networks. *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence* (pp. 139-147). San Francisco: Morgan Kaufmann Publishers.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. In C. Chatfield, M. Tanner & J. Zidek (Eds.), *Texts in statistical science series*, 2nd edition. Chapman & Hall/CRC.

Golightly, A., & Wilkinson, D. J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, *52*, 1674–1693. doi:10.1016/j.csda.2007.05.019

Goutsias, J., & Kim, S. (2006). Stochastic transcriptional regulatory systems with time delays: A mean field approximation. *Journal of Computational Biology*, *13*(5), 1049–1076. doi:10.1089/cmb.2006.13.1049

Gouze, J.-L. (1998). Positive and negative circuits in dynamical systems. *Journal of Biological System*, *6*(21), 11–15. doi:10.1142/S0218339098000054

Gustafsson, M., Hörnquist, M., & Lombardi, A. (2005). Constructing and analyzing a large-scale gene-to-gene regulatory network–lasso constrained inference and biological validation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*(3), 254–261. doi:10.1109/TCBB.2005.35

Guthke, R., Möller, U., Hoffmann, M., Thies, F., & Töpfer, S. (2005). Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection. *Bioinformatics (Oxford, England)*, *21*(8), 1626–1634. doi:10.1093/bioinformatics/bti226

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2002). Combining location and expression data for principled discovery of genetic network models. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *7*, 437–449.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics (Oxford, England)*, *19*(17), 2271–2282. doi:10.1093/bioinformatics/btg313

Imoto, S., Higuchi, T., Goto, T., Kuhara, S., & Miyano, S. (2003). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Proceedings IEEE Computer Society Bioinformatics Conference, (CSB'03)* (pp. 104-113).

Kass, R. E., & Wassermann, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association, 91,* 1343–1370. doi:10.2307/2291752

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4593), 671–680. doi:10.1126/science.220.4598.671

Kitano, H. (2002). Systems biology: A brief overview. *Science, 295,* 1662–1664. doi:10.1126/science.1069492

Kitano, H. (2007). Towards a theory of biological robustness. *Molecular Systems Biology, 3*(137), 1–7.

Kloster, M., Tang, C., & Wingreen, N. S. (2005). Finding regulatory modules through large-scale gene expression analysis. *Bioinformatics (Oxford, England), 21*(7), 1172–1179. doi:10.1093/bioinformatics/bti096

Lavine, M. L. (1999). What is Bayesian statistics and why everything else is wrong. *Journal of Undergraduate Mathematics and Its Applications, 20,* 165–174.

Li, F., Long, T., Lu, Y., Ouyangm, Q., & Tang, C. (2004). The yeast cell cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America, 101,* 4781–4786. doi:10.1073/pnas.0305937101

Li, Y., Campbell, C., & Tipping, M. (2002). Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics (Oxford, England), 18*(10), 1332–1339. doi:10.1093/bioinformatics/18.10.1332

Lu, J., Engl, H. W., & Schuster, P. (2006). Inverse bifurcation analysis: Application to simple gene systems. *Alg. Mol. Biol., 1*(11).

Maritz, J. S., & Lwin, T. (1989). *Empirical Bayes methods*. Chapman & Hall, 2nd edition.

Mestl, T., Plahte, E., & Omholt, S. W. (1995). A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology, 176,* 291–300. doi:10.1006/jtbi.1995.0199

Michaelis, L., & Menten, M. (1913). Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift, 49,* 333–369.

Murphy, K., & Mian, S. (1999). *Modelling gene expression data using dynamic Bayesian networks* (Tech. Rep.). Computer Science Division, University of California, Berkeley, CA.

Ozbudak, E. M., Thattai, M., Kurtser, I., Grossman, A. D., & Oudenaarden, A. v. (2002). Regulation of noise in gene expression of a single gene. *Nature Genetics, 31,* 69–73. doi:10.1038/ng869

Pe'er, D. (2005). Bayesian network analysis of signaling networks: A primer. *Science's STKE, 281,* 14.

Pe'er, D., Regev, A., Elidan, G., & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics (Oxford, England)*, *17*, 215–225.

Peifer, M., & Timmer, J. (2007). Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. *IET Systems Biology*, *1*(2), 78–88. doi:10.1049/iet-syb:20060067

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2002). *Numerical recipes in C++*. Cambridge: Cambridge University Press.

Radde, N., & Kaderali, L. (2007). Bayesian inference of gene regulatory networks using gene expression data. (LNBI 4414), Bird 07, Springer series, 1-15.

Raser, O. M., & O'Shea, E. K. (2004). Control of stochasticity in eukaryotic gene expression. *Science*, *304*, 1811–1814. doi:10.1126/science.1098641

Rechenberg, I. (1973). *Evolutionsstrategie-optimierung technischer systeme nach prinzipien der biologischen evolution*. Unpublished doctoral dissertation. Reprinted by Frommann-Holzboog Verlag, Stuttgart-Bad Cannstatt.

Rogers, S., & Girolami, M. (2005). A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics (Oxford, England)*, *21*(14), 3131–3137. doi:10.1093/bioinformatics/bti487

Sabatti, C., & James, G. M. (2006). Bayesian sparse hidden components analysis for transcription regulation networks. *Bioinformatics (Oxford, England)*, *22*(6), 739–746. doi:10.1093/bioinformatics/btk017

Sanguinetti, G., Lawrence, N. D., & Rattray, M. (2006). Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities. *Bioinformatics (Oxford, England)*, *22*(22), 2775–2781. doi:10.1093/bioinformatics/btl473

Savageau, M., & Alves, R. (2006, July 31- August 4). Mathematical representation and controlled comparison of biochemical systems. Tutorial at the *International Conference on Molecular Systems Biology (ICMSB)*, Munich, Germany.

Spellman, P. T., Sherlock, G., & Zhang, M. Q. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, *9*, 3273–3297.

Srivastava, R., You, L., Summer, J., & Yin, J. (2002). Stochastic versus deterministic modeling of intracellular viral kinetics. *Journal of Theoretical Biology*, *218*(3), 309–321. doi:10.1006/jtbi.2002.3078

Strogatz, S. H. (Ed.). (2000). *Nonlinear dynamics and chaos. Studies in nonlinearity*. Westview Press.

Thieffry, D. (2007). Dynamical roles of biological regulatory circuits. *Briefings in Bioinformatics*, *8*(4), 220–225. doi:10.1093/bib/bbm028

Thomas, R. (1998). Laws for the dynamics of regulatory networks. *J. Dev. Biol.*, *42*, 479–485.

Thomas, R., & D'Ari, R. (1990). *Biological feedback*. Boca Raton, FL: CRC Press.

Thomas, R., Thieffry, D., & Kauffman, M. (1995). Dynamical behaviour of biological regulatory networks–biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, *57*, 247–276.

Vallabhajosyula, R. R., Chickarmane, V., & Sauro, H. M. (2006). Conservation analysis of large biochemical networks. *Bioinformatics (Oxford, England)*, *22*(3), 346–353. doi:10.1093/bioinformatics/bti800

van Someren, E. P., Vaes, B. L. T., Steegenga, W. T., Sijbers, A. M., Dechering, K. J., & Reinders, M. J. T. (2006). Least absolute regression network analysis of the murine osteoblast differentation network. *Bioinformatics (Oxford, England)*, *22*(4), 477–484. doi:10.1093/bioinformatics/bti816

Werhli, A. V., & Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, *6*(1). doi:10.2202/1544-6115.1282

Yagil, G., & Yagil, E. (1971). On the relation between effector concentration and the rate of induced enzyme synthesis. *J. Biophys.*, *11*(1), 11–27. doi:10.1016/S0006-3495(71)86192-1

Zou, M., & Conzen, S. D. (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics (Oxford, England)*, *21*(1), 71–79. doi:10.1093/bioinformatics/bth463

ADDITIONAL READING

De Hoon, M. J. L., Imoto, S., Kobayashi, K., Ogasawara, N., & Miyano, S. (2003). Inferring gene regulatory networks from time-ordered gene expression data of *Bacillus subtilis* using differential equations. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *4*, 29–40.

De Hoon, M. J. L., Makita, Y., Imoto, S., Kobayashi, K., Ogasawara, N., Nakai, K., & Miyano, S. (2004). Predicting gene regulation by sigma factors in *Bacillus subtilis* from genomewide data. *Bioinformatics (Oxford, England)*, *20*(Suppl. 1), i101–i108. doi:10.1093/bioinformatics/bth927

De Jong. (2002) and Kaderali & Radde (2007) provide a general overview of commonly used model classes for biological networks and related inference approaches. An introduction to differential equation models based on chemical reaction kinetics similar to the model used here can be found in Alon (2006). A comprehensive introduction to Bayesian data analysis is given in Gelman et al. (2003). Dynamic Bayesian networks are discussed in Dojer et al. (2006). Some relations between Bayesian networks and stochastic differential equations are pointed out in de Hoon et al. (2003) and de Hoon et al. (2004). The book of Wilkinson (2006) gives an introduction to stochastic kinetic models in systems biology. Further details about the application results presented are given in Kaderali & Radde (2007) and Radde & Kaderali (2007). Alon, U. (2006). *An introduction to systems biology-design principles of biological circuits*. Mathematical and Computational Biology series. London: Chapman & Hall/CRC.

De Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*(1), 67–103. doi:10.1089/10665270252833208

Dojer, N., Gambin, A., Mizera, A., Wilczynski, B., & Tiuryn, J. (2006). Applying dynamic Bayesian networks to perturbed gene expression data. *BMC Bioinformatics*, *7*(249).

Friedman, N., Murphy, K., & Russell, S. (1998). Learning the structure of dynamical probabilistic networks. In *Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence* (pp. 139-147). San Francisco: Morgan Kaufmann Publishers.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. In C. Chatfield, M. Tanner & J. Zidek (Eds.), *Texts in statistical science series*, 2nd edition. Chapman & Hall/CRC.

Kaderali, L., & Radde, N. (2007). Inferring gene regulatory networks from gene expression data. In *Computational Intelligence in Bioinformatics*. Berlin: Springer.

Radde, N., & Kaderali, L. (2007). Bayesian inference of gene regulatory networks using gene expression data. (LNBI 4414), Bird 07, Springer series, 1-15.

Wilkinson, D. J. (2006). Stochastic modelling for systems biology. In *Mathematical and computational biology series*, 1st edition. Chapman & Hall/CRC.

KEY TERMS AND DEFINITIONS

Gene Regulatory Network (GRN): Here a directed graph G(V,E) with n nodes corresponding to n genes in the network. An edge from node j to node i indicates that gene product j has an influence on the expression rate of gene i. This influence is assumed to be either activating or inhibiting. The dynamics of the system is described by ordinary differential equations.

Regulation Function ri: Coupling term in the differential equations for GRNs. Function that describes the influence of regulators on the expression rate of gene i. For simplicity, it is often assumed that different regulators act independently, and their influences can be decoupled. An influence of a single regulator j on i is then often described with simple linear functions, or Michaelis-Menten and Hill equations are used, which can be derived from chemical reaction kinetics.

Quasi-Steady State Approximation (QSSA): A method to reduce the number of variables of a system that includes processes on different time scales which can be separated into slow and fast. One assumes that the fast processes are always in a steady state, which changes on the slow time scale. For GRNs, the fast time scale corresponds to transcription factor – DNA binding, and the relevant slow time scale is given by the expression rates. Here, the QSSA allows for a functional relation between gene product levels and their effect on the expression rates of regulated genes, as it is implicitly assumed in most network inference approaches.

Stochastic Modeling Approach: In stochastic modeling approaches for GRNs, observed gene expression values are interpreted as random variables, and the network inference problem translates into characterizing their probability distributions from measurements. Contrary to deterministic models, these approaches can capture variability across different cells or experiments. Different stochastic approaches have been introduced, and here we suggest a classification according to whether the system itself is stochastic or noise stems from the measurement process.

Bayes Regularized Differential Equation: Specific stochastic modeling approach in which the noise is assumed to stem solely from the measurement process. The state of the system is deterministic and uniquely determined by a differential equation. Observations that are used for network inference are random variables due to measurement noise. This allows for a Bayesian regularization for network inference.

Section 3

Modeling Methods

Chapter 7
Computational Approaches for Modeling Intrinsic Noise and Delays in Genetic Regulatory Networks

Manuel Barrio
University of Valladolid, Spain

Kevin Burrage
The University of Oxford, UK

Pamela Burrage
The University of Queensland, Australia

André Leier
ETH Zurich, Switzerland

Tatiana Márquez Lago
ETH Zurich, Switzerland

ABSTRACT

This chapter focuses on the interactions and roles between delays and intrinsic noise effects within cellular pathways and regulatory networks. We address these aspects by focusing on genetic regulatory networks that share a common network motif, namely the negative feedback loop, leading to oscillatory gene expression and protein levels. In this context, we discuss computational simulation algorithms for addressing the interplay of delays and noise within the signaling pathways based on biological data. We address implementational issues associated with efficiency and robustness. In a molecular biology setting we present two case studies of temporal models for the Hes1 gene (Monk, 2003; Hirata et al., 2002), known to act as a molecular clock, and the Her1/Her7 regulatory system controlling the periodic somite segmentation in vertebrate embryos (Giudicelli and Lewis, 2004; Horikawa et al., 2006).

DOI: 10.4018/978-1-60566-685-3.ch007

1. INTRODUCTION

The mathematical modeling and simulation of genetic regulatory networks can provide insights into the complicated biological and chemical processes associated with genetic regulation. However, highly resolved computational models of such biochemical complexity can be very expensive and often infeasible and, thus, it is important that the models are kept simple but nevertheless capture the key processes.

Two vital aspects in modeling genetic regulatory networks are intrinsic noise and delays. Intrinsic noise arises in the system when there are small to moderate numbers of certain key molecules and is due to the uncertainty of knowing when a reaction occurs and which reaction it might be. Intrinsic noise is entirely different to extrinsic noise in which state changes are due to fluctuations in external conditions, such as temperature. These intrinsic noise effects can be modeled through the Stochastic Simulation Algorithm (SSA), first applied by Gillespie (1977) to simulate discrete chemical kinetics as the evolution of a discrete nonlinear Markov process.

Delays are intrinsic to slow biochemical processes that do not occur instantaneously and are often affected by spatial inhomogeneities. For instance, they are often associated with transcription and translation, two processes that imply other spatiotemporal processes often not explicitly modeled, such as (in eukaryotes) diffusion and translocation into and out of the nucleus, RNA polymerase activation, splicing, protein synthesis, and protein folding. These processes can take many minutes and so the effects are very important especially in the laying down of oscillating patterns of gene expression (Hirata et al., 2002). Monk (2003) notes that in mouse there is an average delay of 10–20 minutes between the action of a transcription factor on the promoter region of a gene and the appearance of the corresponding mRNA in the cytosol. Similarly, there is a delay of typically 1–3 minutes for the translation of a protein from mRNA.

By incorporating delays into the temporal model we can capture essential information on a macroscopic level, the delay can itself account for the multitude of biochemical processes and events on a microscopic time scale that render us unable to compute cell dynamics in real-time. Hence, we can expect more accurate and reliable predictions of cellular dynamics through the use of time delay models (Barrio et al., 2006).

One of the first people to consider feedback differential equation models for the regulation of enzyme synthesis was Goodwin (1965). An der Heiden (1979) then modified these ideas by including transport delays into Goodwin's model. The oscillatory behavior of the ensuing delay differential equations (DDEs) as a function of the size of delays was investigated by an der Heiden. However, these DDE models act in the continuous deterministic regime and this regime is not always appropriate when considering small numbers of molecules such as in the case of genetic regulation with small numbers of transcription factors.

In a lovely set of experiments, Hirata et al. (2002) measured the production of *hes1* mRNA and Hes1 protein in mice. This work forms the basis of one of our case studies in Section 4.1. Serum treatments on cultured cells result in oscillations in expression levels for *hes1* mRNA and Hes1 protein in a two hour cycle with a phase lag of approximately 15 minutes between the oscillatory profiles of mRNA and protein. The oscillations in expression continue for 6 to 12 hours.

In order to explain the observed behaviors, Hirata et al. modified a mathematical model developed by Elowitz and Leibler (2000) for a synthetic gene network constructed in *E. coli* cells by introducing one gene from λ-phage. By postulating a Hes1 interacting factor as a third molecular species Hirata et al. obtained a system of three Ordinary Differential Equations (ODEs) that gives rise to sustained

oscillatory behavior. However, there is no direct experimental evidence for such an interacting factor. Rather, the introduction of a third variable is due to the fact that certain systems of two ODEs cannot generate sustained oscillations. This observation together with the experimental results of Hirata et al. led to a number of papers in which simple coupled delay differential equations were developed in order to explain the sustained oscillations without recourse to the addition of a third variable (Monk, 2003; Jensen et al., 2003; Lewis, 2003; Bernard et al., 2006).

Barrio et al. (2006) took a different approach from the above authors and tried to explain the results of Hirata et al. by taking proper account of both time delays and intrinsic randomness. They developed a Delay Stochastic Simulation Algorithm (DSSA) that generalizes the Stochastic Simulation Algorithm (SSA) to the delayed setting. Independently, Bratsun et al. (2005) developed a delay SSA without considering waiting times for delayed reactions while only non-consuming reactions can be specified to be delayed. More recently, Cai (2007) introduced a direct delay SSA method and showed that both, the DSSA by Barrio et al. and the direct method are exact stochastic simulation algorithms for chemical reaction systems with delays. The experimental results of Hirata et al. seemed to be better explained through the delay stochastic simulation algorithm approach rather than through delay differential equations (Barrio et al., 2006).

When modeling biological systems with large numbers of molecules and/or rate constants, the time steps in stochastic simulation algorithms can become very small and, hence, the simulation can be computationally highly expensive. By consequence, this limits the feasible 'real-time' span of the simulations. In order to reduce the computational load we need new algorithms that still model intrinsic noise in a delayed setting but overcome the issues of small step sizes. Temporal coarse-graining has been considered through the use of τ -leap methods (Gillespie, 2001; Tian and Burrage, 2004; Peng et al. 2007, Anderson, 2007, 2008), and similar ideas have been applied in the delay setting (Leier et al., 2008(a)), thus rendering an efficient algorithm that yields accurate simulations in time spans that are long enough to be of actual interest to the experimentalists.

Lastly, temporal delay models lack spatial resolution but nevertheless allow for portraying spatial aspects of cellular processes by compartmentalization, that is, by distinguishing between identical molecular species according to their location. Recent research suggests that molecular translocation processes can be well captured and modeled by means of time delayed processes with specific delay distributions. However, it is worth mentioning that spatial algorithms are not replaceable in all cases. Examples of the latter are scenarios with high spatial heterogeneity, anisotropies, or when single-particle tracking becomes strictly necessary. Spatial highly-resolved algorithms are computationally most expensive, and coarse-graining techniques have also been developed for this case (Chatterjee and Vlachos, 2005; Chatterjee and Vlachos, 2006).

The outline of this Chapter is as follows. In section 2 we give an overview of some of the approaches to the temporal modeling of chemical kinetics. In section 3 we present various types of simulation algorithms with and without delays and discuss how we can improve the accuracy and robustness by so-called τ leap approaches. Section 4 gives two case studies: the Hes1 molecular clock and the Her1/7 complex which plays a role in somite formation in zebrafish. Section 5 presents some conclusions.

2. MODELING CHEMICAL KINETICS

Modeling and simulations are valuable tools for investigating complex biochemical systems. Not only do they allow us to determine if a proposed reaction mechanism is consistent with observed experimental results, but they can also aid experimental design techniques by exploring reaction network interactions with relative ease. The choice for a particular modeling approach depends on several factors, such as molecular concentrations, distributions, the type of reactions and their time scales, whether discreteness and internal noise have noticeable macroscopic effects and, lastly, if the model requires spatial information.

Deterministic models assume a time evolution that is both continuous and predictable. However, randomness is intrinsic to biological systems, where system behavior is typically represented by noisy signals. Often the most important source of stochasticity stems from the fact that molecular reactions are random events, as it is impossible to say with certainty the specific type of reaction that will happen next, or when or where such event is to occur. Moreover, low molecular concentrations, coupled to random diffusion, are an important source of spatial inhomogeneity and stochastic variation.

In a purely temporal setting, and when there are large numbers of molecules present, chemical reactions are modeled by ordinary differential equations that are based on the laws of Mass Action and the fact that reaction rates can be estimated on the basis of average values of the reactant density. Any set of *m* chemical reactions can be characterized by two sets of quantities: the stoichiometric vectors (update rules for each reaction) $\nu_1,...,\nu_m$ and the propensity functions $a_1(X(t)),...,a_m(X(t))$. The propensity functions represent the relative probabilities of each of the *m* reactions occurring. Here X(t) is the vector of concentrations at time *t* of the *N* species involved in the reactions. The ODE that describes this chemical system, under the Law of Mass Action, is given by

$$X'(t) = \sum_{j=1}^{m} \nu_j a_j(X(t)). \tag{1}$$

In order to make this clearer we give a simple example for Michaelis–Menten kinetics. This system involves a substrate (S), an enzyme (E), a complex (C) and a product (P). The kinetics can be written as

$$E + S \longrightarrow C,$$

$$C \longrightarrow E + S,$$

$$C \longrightarrow E + P.$$

Let X(t) be the concentration of (E(t), S(t), C(t)) then the stoichiometric vectors (or the update rules for each of the three reactions) are

$$\nu_1 = (-1,-1,1)^T, \quad \upsilon_2 = (1,1,-1)^T, \quad \nu_3 = (1,0,-1)^T.$$

The time dependent propensity functions $a_1(X(t)),...,a_m(X(t))$ are the relative probabilities of each of the three reactions occurring, respectively, and are given by

$$a_1(X) = k_1 ES$$
$$a_2(X) = k_2 C$$
$$a_3(X) = k_3 C.$$

In this case (1) becomes

$$X_1^{'} = -k_1 X_1 X_2 + (k_2 + k_3) X_3$$
$$X_2^{'} = -k_1 X_1 X_2 + k_2 X_3$$
$$X_3^{'} = k_1 X_1 X_2 - (k_2 + k_3) X_3.$$

Often in such systems there is a conservation of molecular numbers (here $X_1{}' + X_3{}' = 0$) and so one or more equations can be removed. Additional equations can be removed by the use of the Quasi-Steady State Assumption (QSSA). Under the QSSA it is assumed that the fast reactions go to equilibrium much more quickly that the slow reactions. Thus a system of algebraic equations can be solved at the "fast equilibrium" and this solution substituted back into the original system, thus reducing the dimension and altering the propensity functions to include nonlinear Hill functions.

In the case of small numbers of molecules the appropriate modeling formulation is the Stochastic Simulation Algorithm, as ODEs can only describe a mean behavior. The SSA is essentially an exact procedure that describes the evolution of a discrete nonlinear Markov process. It accounts for the inherent stochasticity (internal noise) of the *m* reacting channels and only assigns integer numbers of molecules to the state vector. At each step, the SSA simulates two random numbers (representing probabilities) from the uniform distribution U[0,1] to evaluate an exponential waiting time, τ, for the next reaction to occur and an integer *j* between 1 and *m* that indicates which reaction occurs. The state vector is updated at the new time point by the addition of the j^{th} stoichiometric vector to the previous value of the state vector, that is

$$X(t+\tau) = X(t) + \nu_j.$$

The main limiting feature of SSA is that the time step can become very small, especially if there are large numbers of molecules or widely varying rate constants. In order to overcome these limitations, a number of different approaches (so called τ-leap methods) have been suggested in which the sampling of likely reactions is taken from either Poisson (Gillespie, 2001) or Binomial (Tian and Burrage, 2004) distributions. In these cases a much larger time step can be used at the loss of a small amount of accuracy. Cao et al. (2006) have analyzed effective strategies for choosing the step size in τ-leap methods. The reason sampling occurs from a Poisson distribution is due to the fact that the SSA can also be viewed as a type of τ leap method based on Poisson sampling (Kurtz, 1971). On the other hand, Binomial sampling is valid because as the number of molecules becomes large, Poisson random variables are well approximated by Binomial random variables.

A very different approach is to note that the discrete nonlinear Markov process described by the SSA has a probability density functions that is the solution of the so-called Chemical Master Equation

(CME). The CME is a discrete parabolic partial differential equation in which there is an equation for each configuration of the State Space. When the State Space is enumerated, the CME becomes a linear ODE and the probability density function takes the form

$$p(t) = e^{At}p(0)$$

where *A* is the state-space matrix. Even for relatively small systems, the dimension of *A* can be in the millions, so it would appear that this is not a computationally feasible approach. However, one should consider that not all of the states are reachable. Furthermore, a proposed finite state projection algorithm (Munsky and Khammash, 2006) reduces the size of the matrix *A*. Then one can use Krylov subspace techniques (Burrage et al., 2006) to efficiently compute the exponential of a matrix times a vector, making the computation of the probability density function directly a very feasible technique (MacNamara et al., 2007).

Finally, it is important to note that there is a regime intermediate to the discrete stochastic regime and the continuous deterministic ODE regime in which the internal noise effects are still significant but continuity arguments can apply. This leads to the so-called Chemical Langevin Equation (CLE) that is an Itô stochastic ordinary differential equation (SDE), driven by a set of Wiener processes that describes the fluctuation in the concentrations of the molecular species. The CLE preserves the correct dynamics for the first two moments of the SSA and takes the form

$$dX = \sum_{j=1}^{m} \nu_j a_j(X(t)) + B(X(t))dW(t).$$

Here $W(t) = (W_1(t),...,W_N(t))$ is a vector of N independent Wiener processes whose increments $\Delta W_j = W_j(t+h) - W_j(t)$ are $N(0,h)$ and where

$$B(x) = \sqrt{C}, \quad C = (\nu_1,...,\nu_m)Diag(a_1(X),...,a_m(X))(\nu_1,...,\nu_m)^T.$$

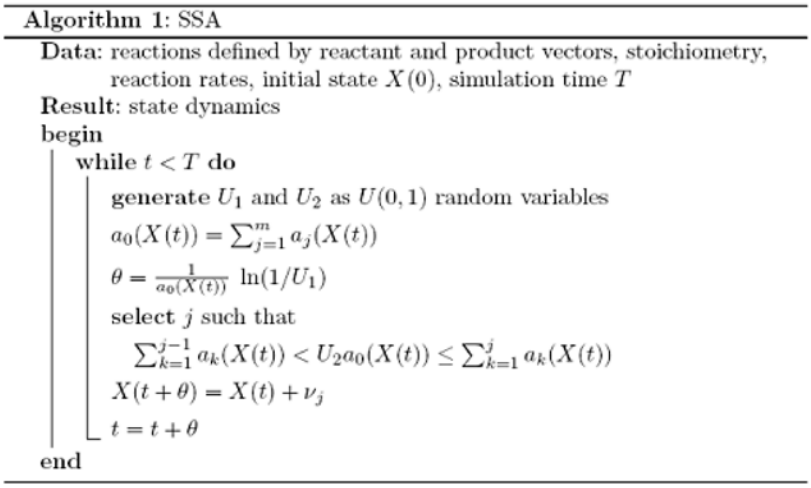
Here *h* is the time discretization step. This formulation can be derived from the Poisson formulation of the SSA by noting that as $Th \to \infty$ with $h \to 0$,

$$\begin{aligned} P(Th) \to N(Th,Th) &= Th + \sqrt{Th}\, N(0,1) \\ &= Th + \sqrt{T}\, \Delta W. \end{aligned}$$

Effective numerical methods designed for the numerical solution of SDEs (such as the Euler-Maruyama method) can be used to simulate the chemical kinetics in this intermediate regime. Furthermore, adaptive multiscale methods have been developed which attempt to move back and forth between these three regimes as the numbers of molecules change (Burrage et al., 2004).

None of these frameworks explicitly incorporate delay affects but in fact the same modeling regimes arise in a natural fashion if delay is included. These have been thoroughly explored in Barrio et al. (2006)

Figure 1. The Stochastic Simulation Algorithm



and Tian et al. (2007) in terms of the same modeling regimes mentioned above. We now discuss some of the issues when incorporating noise and delays.

3. SIMULATION ALGORITHMS

In recent years, discrete stochastic simulation techniques have been widely used to help understand the dynamic behavior of biochemical systems such as genetic regulatory networks and intra-cellular and inter-cellular signaling pathways when there are small to moderate numbers of molecular species involved. In addition to the methods mentioned above, other simulation type methods have also been proposed recently, for example, Gibson and Bruck's next reaction method (2000), Gillespie's continuous model (2000) and the probability-weighted Monte-Carlo approach by Resat et al. (2001). In this section we review some of these approaches without and with delays and then discuss extensions via tau leaping strategies which can dramatically improve robustness and computational performance.

3.1 SSA

The SSA (Stochastic Simulation Algorithm) in Figure 1 is a numerical Monte Carlo procedure that can be used to simulate the time evolution of a set of molecular species affected by a given set of reactions. It was introduced by Gillespie (1977) as an exact calculation that generates simulated trajectories of the system state. These trajectories are numerical realizations of the Chemical Master Equation (CME). It is important to note that the SSA is based on a fundamental stochastic premise that defines the probability, given a particular state that one reaction will occur in the next infinitesimal time internal. This assumption is used without approximation by the SSA and makes it exact with respect to the CME.

More precisely, consider a well-stirred volume Ω of molecules containing N molecular species $\{S_1,...,S_N\}$ that interact at constant temperature through M chemical reactions $\{R_1,...,R_M\}$. Given the

system state at a particular time **X**(t) which represents the number of molecules of each species, we can define for each reaction R_j (j=1,...,M) its propensity function $a_j(\mathbf{x})$ in a given state **X**(t)=**x** so that

$a_j(\mathbf{x})dt$ = probability that one R_j reaction will occur somewhere inside Ω in the next infinitesimal time interval [t,t+dt).

Additionally, each reaction is characterized by its stoichiometric vector ν_j that defines the state change in the number of species due to reaction R_j.

The procedure to generate simulated trajectories of **X**(t) is based on the probability function of the two random variables: (1) the time τ to the next occurring reaction, and (2) the index j of the next reaction. Given a current state **x**, the probability of state change per unit of time is constant ($a_0(\mathbf{x})$) and so the waiting time to the next reaction is an exponential random variable with mean $1/a_0(\mathbf{x})$. The reaction index j is an integer random variable with point probabilities

$a_j(\mathbf{x}) / a_0(\mathbf{x})$, where $a_0(\mathbf{x}) = \sum_{k=1}^{M} a_k(\mathbf{x})$.

These two random variables and their distributions are the basis of the SSA. One of the simplest Monte Carlo procedures for generating time and index of the next reaction is the so-called 'direct method'. Two independent random numbers r_1 and r_2 are drawn from the uniform distribution in the unit interval U(0,1), and then τ is assigned as

$$\tau = \frac{1}{a_0(x)} \ln\left(1/r_1\right),$$

while j is the reaction index that satisfies

$$\sum_{k=1}^{j-1} a_k(x) < r_2 \cdot a_0(x) \leq \sum_{k=j}^{j} a_k(x).$$

Then the system is updated by $\mathbf{x}(t+\tau) = \mathbf{x}(t) + \nu_j$, and the procedure is repeated to evolve the system through time. Figure 2 is an algorithmic representation of the direct method.

3.2 Delay SSA

Biological processes often involve complex reactions and mechanisms that cannot be considered instantaneous. Reactants are processed and products are not present until a certain future time point. This time delay should be incorporated into our computational models if we want to capture a faithful representation of the biological process. Additionally, delays are often important parameters that affect the dynamic evolution of the system. A system of DDEs can take the general form

$$y' = f(t, y(t), y(t-\tau)),$$

and in the case of chemical kinetics as described by (1), the DDE formulation is

Figure 2.

Algorithm 2: DSSA

Data: reactions defined by reactant and product vectors, consuming delayed reactions are marked, stoichiometry, reaction rates, initial state $X(0)$, simulation time T, delays
Result: state dynamics
begin
  while $t < T$ **do**
    generate U_1 and U_2 as $U(0,1)$ random variables
    $a_0(X(t)) = \sum_{j=1}^{m} a_j(X(t))$
    $\theta = \frac{1}{a_0(X(t))} \ln(1/U_1)$
    select j such that
      $\sum_{k=1}^{j-1} a_k(X(t)) < U_2 a_0(X(t)) \leq \sum_{k=1}^{j} a_k(X(t))$
    if *delayed reactions are scheduled within* $(t, t+\theta]$ **then**
      let k *be the delayed reaction scheduled next at time* $t+\tau$
      if k *is a consuming delayed reaction* **then**
        $X(t+\tau) = X(t) + \nu_k^p$ (update products only)
      else
        $X(t+\tau) = X(t) + \nu_k$
      $t = t + \tau$
    else
      if j *is not a delayed reaction* **then**
        $X(t+\theta) = X(t) + \nu_j$
      else
        record time $t + \theta + \tau_j$ for delayed reaction j with delay τ_j
        if j *is a consuming delayed reaction* **then**
          $X(t+\theta) = X(t) + \nu_j^s$ (update reactants)
      $t = t + \theta$
end

$$X'(t) = \sum_{j=1}^{m} \upsilon_j a_j \left(X\left(t - \tau_j\right)\right).$$

There are a number of suitable numerical methods for solving such systems, some of which are implemented in MATLAB. However, if intrinsic noise is important then we need a generalization of the *stochastic simulation algorithm* (SSA) for chemical kinetics with delayed reactions. The DSSA differs from the SSA by making a clear distinction between the reaction waiting time and reaction delay. The former is the time between two consecutive reactions whereas the latter is the time elapsed from the processing of the reactants to the appearance of the products.

Simulation proceeds in the standard way (SSA) if non-delayed reactions take place. However, if the next reaction index points to a delayed reaction then we have to distinguish between two different types: consuming and non-consuming. In case of non-consuming reactions, the corresponding reactants and products are not updated. Instead, the state update is scheduled for ‘present time + delay’ which will be

reached in a future simulation step. When that happens, the last drawn reaction is ignored and instead the state is updated according to the delayed reaction. Simulation continues at the delayed reaction time point. On the other hand, if the reaction is consuming, reactants and products of delayed consuming reactions must be updated separately: (1) reactant consumption updates the state when the delayed reaction is selected and (2) product generation is updated when the reaction is completed.

The trajectories simulated by SSA are numerical realizations of the state evolution $\mathbf{X}(t)$. Additionally, the probability density function of $\mathbf{X}(t)$ is completely determined by the Chemical Master Equation. Similarly, a CME for the DSSA, namely a DCME, has been derived from first principles and the DSSA has a corresponding representation as a system of delay differential equations (DDEs) – see Barrio et al. (2006) and Tian et al. (2008).

Figure 2 is an algorithmic description of the DSSA dealing with both delayed and non-delayed, as well as with consuming and non-consuming, reactions. Time steps are defined either by a next reaction waiting time or by a delayed time update.

3.3 Spatial Methods

In many Cell Biology settings spatially resolved simulations are mandatory. Some common examples in which spatial simulations are unavoidable are systems embedded in complex spatial structures, molecular motion described by low diffusion rates, or systems containing significantly low numbers of molecules, to name a few. The most straightforward spatial technique is through reaction-diffusion partial differential equations. However, this approach is only valid if dealing with large molecular concentrations and when noise is not amplified throughout the system. If at least one of these conditions fails to hold, one must rely on spatial stochastic simulators, which can be discrete or continuous in nature and have different levels of spatial resolution.

It should always be kept in mind that there is a trade-off between simulation time and resolution. That is, the more highly-resolved, the more computationally expensive these simulations become. The highly resolved end of the discrete spatial stochastic simulators spectrum is represented by lattice and off-lattice particle based methods. In lattice methods a two-dimensional or three-dimensional computational lattice is used to represent a membrane or the interior of some part of a cell (Turner et al., 2004; Morton-Firth and Bray, 1998; Nicolau et al., 2006). Such a lattice is then "populated" with particles of different molecular species that may diffuse throughout the simulation domain by jumping to empty neighboring sites and, depending on user-specified reaction rules, interacting chemically with a certain probability. Such lattice-based simulators are commonly referred to as Kinetic Monte Carlo Methods.

In off-lattice methods, particles have their own specific spatial coordinates and reaction bins whose size depends on the particular diffusion rates are drawn around them. If one or more molecules happen to be inside such a bin, appropriate chemical reactions can take place with a certain probability, and if a reaction is readily performed, the reactant particles are flagged. It should be noted that in off-lattice methods, the domains and/or compartments are usually still discretized to efficiently localize particles.

Particle methods can provide very detailed simulations of highly complex systems at the cost of exceedingly large amounts of computational time and, possibly, restrictions on the size of the simulation domain. Hence, such detailed simulations can often only yield short simulation time spans that may not be of sufficient interest to experimentalists.

3.4 Coarse Grained Methods

A major drawback of delayed and non-delayed, spatial and non-spatial stochastic simulation algorithms are their high computational costs when dealing with large numbers of molecules or widely varying rate constants. These factors inevitably result in exceedingly small simulation time steps, making the overall simulation computationally expensive or even infeasible. In order to reduce the computational load, we can coarsen the simulation, accounting for many events in one single larger time step. This is the general idea behind the so-called *τ*-leap methods, where the simulation advances in time leaps while updating the system state according to a reasonably good approximation for the accumulated number of reactions (and diffusions if a spatial simulation) within the time step.

3.4.1 τ-leap Methods

Gillespie (2001) proposed the *Poisson τ-leap method* in which the number of reactions in each *τ*-leap are sampled from a Poisson distribution, and the *τ* step is controlled by a selection strategy that depends on a pre-specified control parameter ε, such that $0<\varepsilon<<1$.

The update procedure for the Poisson *τ*-leap method can be written as $x\left(t+\tau\right)=x\left(t\right)+\sum_{j=1}^{M}K_j\nu_j$, where $K_j=P\left(a_j\left(X\left(t\right)\right)\tau\right)$, for reactions $j=1,\ldots,M$, is a sample from the Poisson distribution with mean $a_j\left(X\left(t\right)\right)\tau$. Further improvements were made by Gillespie and Petzold (2003), Rathinam *et al.* (2003), and Cao *et al.* (2005, 2006).

However, samples from a Poisson distribution range from zero to unbounded values. Hence, when updating the system, negative numbers of molecules can occur if larger step sizes are used. In order to avoid this, Tian and Burrage (2004) and later Chatterjee *et al.* (2006) proposed the Binomial *τ*-leap method where the numbers of reactions in a leap are drawn from a Binomial distribution. Thus, the various K_j take the form $K_j=B\left(N_j,P_j\right)$, where there are some subtleties in the form of the N_j and P_j, and such variables N_j and P_j represent the sample size and probability of occurrence of reaction type *j*, respectively. Auger et al. (2006) presented a modification to the original Binomial *τ*-leap method which is a more robust implementation than the original formulation. Furthermore, Anderson (2007, 2008) has shown interesting connections between sampling from the Poisson and Binomial distributions in the context of *τ*-leap methods in both a non-delayed and delayed setting.

Recently, Peng *et al* (2007) developed a modified Binomial *τ*-leap method that estimates the number of reaction products within a *τ*-leap step allowing them to participate in additional reactions in the same leap. However, Leier et al. (2008) show that such an approach may not accurately describe complex dynamics including time delays, and they propose a generalized *τ*-leap method, that is described in more detail in Section 3.4.2. Lastly, *τ*-leap methods can also be extended to the spatially resolved spectrum, where the simulation advances in time leaps that account for several molecular diffusion and reaction events, as shown by Marquez-Lago and Burrage (2007) and described in Section 3.4.3.

3.4.2 Bτ-DSSA

Initial Binomial *τ*-leap algorithms (Tian and Burrage, 2004; Peng et al., 2007) were not able to capture accurately the dynamics of certain chemical kinetics compared to the exact SSA/DSSA approach, due to

Table 1. Some simple reactions R_j and their corresponding propensities a_j, stoichiometric coefficients $\nu_{j,\cdot}$ and maximum number of potential reaction events N_j. Hill functions are often used to describe the regulatory effect of one or more transcription factors on the chemical kinetics. For a Hill function depending on a single transcription factor X_k this results in the propensity $a_j = c_j \cdot f(X_k)$. Calculating the $N_j(x)$ for Hill-type reactions involves some subtlety. For Hill type reactions, Leier et al. (2008) define $N_j(x) = C$ where C is some constant. Simulations show that, unless C is too small (< 10), it has no noticeable effect on the simulation outcome.

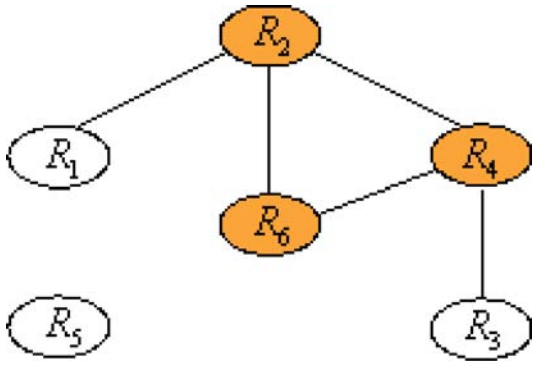
Reaction R_j	Propensity a_j	N_j	Stoichiometric Coefficients
1st order $S_k \xrightarrow{c_j} S_l$	$a_j = c_j \cdot X_k$	X_k	$\nu_{j,k} = -1$, $\nu_{j,l} = 1$
Heterodimeric $S_k + S_l \xrightarrow{c_j} S_m$	$a_j = c_j \cdot X_k X_l$	$\min\{X_k, X_l\}$	$\nu_{j,k} = \nu_{j,l} = -1$, $\nu_{j,m} = 1$
Homodimeric $S_k + S_k \xrightarrow{c_j} S_l$	$a_j = c_j \cdot X_k(X_k - 1)/2$	$\left\lfloor \frac{X_k}{2} \right\rfloor$	$\nu_{j,k} = -2$, $\nu_{j,l} = 1$
Hill type $S_k \xrightarrow{c_j f} S_k + S_l$	$a_j = c_j \cdot f(X_k)$ where (activation) or $f(X_k(t)) = \frac{1}{1 + (X_k(t)/X_0)^h}$ (inhibition) with Hill coefficient *h*.	constant, $N_j >> 1$	$\nu_{j,l} = 1$

insufficient numbers of reactions drawn in τ-leap steps. In Leier et al. (2008) a new generalized Binomial τ-leap method (Bτ-DSSA) is presented that addresses the difficulties associated with complex chemical kinetics and introduces delays into the Binomial τ-leap framework. A description of the Bτ-DSSA is given in Algorithm 3.

Estimating a proper maximum number N_j of potential reaction events of type R_j for the Binomial random variables $B(N_j, P_j)$ is crucial for an accurate reproduction of system dynamics. For specific reactions, Table I shows how to calculate N_j assuming R_j is an isolated reaction that does not share reactants with any other reactions. While this estimation is straightforward for isolated, elementary reaction, it is less obvious for chemical kinetics involving large, interacting reaction networks where multiple reactions share the same reactants.

The Bτ-DSSA samples reaction numbers from Binomial distributions $B(N_j'', P_j)$ (Step 5 in Algorithm 3). Here, $N_j'' = N_j(x, \xi)$, with $\xi \equiv (\xi_1, \ldots, \xi_M)$ and $\xi_i \le N_i(x)$, is the maximal number of potential reaction events of type R_j when $\xi_1, \ldots, \xi_M$ reactions of $R_1, \ldots, R_M$ occur in the τ-step. For $N_j(x, \xi)$ it is assumed that $\xi_j, \ldots, \xi_M = 0$ since only the already sampled reaction numbers $\xi_1, \ldots, \xi_{j-1}$ are considered. However, unlike the original Binomial τ-leap method by Tian and Burrage (2004), the

Figure 3. Artificial chemical kinetics system. The set of reactions R_1 *to* R_6 *constitutes a network where two reactions, i.e. two vertices, are connected by an edge if and only if they have one or more common reactant species. The network has two connected subnetworks,* $\{ R_5 \}$ *and* $\{ R_1, R_2, R_3, R_4, R_6 \}$*. In the original Binomial τ-leap formulation, the maximum number of potential reaction events of type* R_6 *was calculated as the minimum* N_i *(see Table I) over the subnetwork* $\{ R_1, R_2, R_3, R_4, R_6 \}$ *(the subnetwork that* R_1 *belongs to). The Bτ-DSSA calculates* $N_6\left(x, \xi\right)$ *considering only* R_6 *and its direct (shaded) neighbors:* $N_6\left(x, -, \xi_2, -, \xi_4, -, 0\right) = \min\left\{x_2 - \xi_2 - \xi_4, x_5\right\}$ *with* $x_2 = \left[B\right]$ *and* $x_5 = \left[E\right]$.



N_j are calculated considering only those reactions R_i (and hence ξ_i) that share reactant species with R_j. Figure 1 illustrates the difference.

As a consequence, in the Bτ-DSSA the maximal number of potential reaction events is usually larger than in the original Binomial τ-leap method. Numbers of delayed reactions are sampled in the same way as numbers of non-delayed reactions. The update of the system state (Step 6 in Figure 4) has to distinguish between delayed consuming and non-consuming reactions scheduled within the τ-leap, but also has to sample the update times of all delayed reactions drawn for the τ-leap.

Numerical simulations reveal that, unlike previous Binomial τ-leap methods, the Bτ-DSSA is better able to accurately capture the dynamics of oscillating patterns of gene expression. In such systems delayed reactions play a crucial role in maintaining the cyclic behavior and sampling too many or an insufficient number of delayed reactions will inevitably lead to a different cycle frequency. For the applications in Section 4.2, the Bτ-DSSA was able to reproduce the oscillatory dynamics both accurately and significantly faster than the DSSA. In case of the Her1/7-model for 5 coupled cells, Bτ-DSSA was 70 to 100 times faster than the DSSA implementation of Barrio et al. (2006).

$R_1 \colon R_1\ A \longrightarrow B \colon A \longrightarrow B$,

$R_2 \colon R_2\ A + B \longrightarrow C \colon A + B \longrightarrow C$,

$R_3 \colon R_3\ C \longrightarrow A + B \colon C \longrightarrow A + B$,

$R_4 \colon R_4\ B + C \longrightarrow D \colon B + C \longrightarrow D$,

Figure 4.

Algorithm 3: Bτ-DSSA

Data: reactions defined by reactant and product vectors, consuming delayed reactions are marked, stoichiometry: $\nu = -\rho + \pi$ (with ρ and π being update vectors for left-hand-side and right-hand-side of reaction, respectively), reaction rates, initial state $X(0)$, simulation time T, delays, pre-specified $K \in [1, 10]$

Result: state dynamics

begin

1. Calculate $a_1(x), \cdots, a_m(x)$, $a_0(x) = \sum_{j=1}^{m} a_j(x)$; and $N_1(x), \cdots, N_m(x)$

2. Choose a τ-selection procedure: update corresponding variables

3. Check step-size conditions. For each reaction R_j
Calculate $N'_j = \min\{N_i(x), i \in I_j\}$ and $a'_j(x) = \sum_{i \in I_j} a_i(x)$
If $(N'_j > 0)$ AND $(a'_j(x)\tau/N'_j > 1)$ then $\tau = N'_j/a'_j(x)$

4. If $\tau \leq K/a_0$ perform a normal (D)SSA step, otherwise go to (5)

5. Sample. Initialise $\xi \equiv (\xi_1, \cdots, \xi_m) = 0$
For each reaction R_j
Calculate $N''_j = N_j(x, \xi)$
Generate a sample value ξ_j for the reaction number of type R_j
If $N''_j > 0$ then $\xi_j = B(N''_j, P_j)$ with $P_j = a_j(x)\tau/N''_j$ else $\xi_j = 0$.

6. Update non-delayed and delayed reactions. The subscripts nd_j and d_j represent the j^{th} non-delayed and delayed reaction, respectively. M' denotes the number of non-delayed reactions.
6.1 Delayed R_{d_j} with delay δ_j:
Record ξ_{d_j} random update time points $t + \delta_j + u_k\tau$ with $u_k \in U(0, 1), k = 1, \cdots, \xi_{d_j}$
If R_{d_j} is a consuming, delayed reaction, update $x(t + \tau) = x(t + \tau) + \rho_{d_j}$
6.2 Non-delayed: $x(t + \tau) = x(t) + \sum_{j=1}^{M'} \xi_{nd_j}\nu_{nd_j}$
6.3 Delayed (scheduled within $[t, t + \tau)$) :
$x(t + \tau) = x(t + \tau) + \pi_{d_j}$ (consuming, delayed reactions)
$x(t + \tau) = x(t + \tau) + \nu_{d_j}$ (remaining reactions)

end

$R_5 : R_5\ D \longrightarrow B + C : D \longrightarrow B + C$,

$R_6 : R_6\ B + E \longrightarrow F : B + E \longrightarrow F$,

3.4.3 Bτ-SSSA

As mentioned before, particle methods can provide very detailed simulations at the cost of exceedingly large amounts of computational time and, possibly, restrictions on the size of the simulation domain. In other words, we may need to coarsen the simulation in order to provide a spatially resolved method that

yields accurate chemical kinetics in meaningful simulation times that are of actual biological interest to experimentalists.

The idea behind τ-leaping in space is to account for several diffusion and reaction events in one larger time step, without compromising spatial nor temporal accuracy. Marquez-Lago and Burrage (2007) presented the Binomial τ-leap Spatial Simulation Algorithm, Bτ-SSSA, a coarse-grained version of an existing spatial stochastic simulation algorithm known as the next subvolume method (Elf and Ehrenberg, 2004; Elf et al., 2003; Hattne et al. 2005).

The next subvolume method is a generalization of the SSA, where the volume is divided into separate subvolumes that are small enough to be considered homogeneous by diffusion over the time scale of the reaction. At each step, the state of the system is updated by performing an appropriate reaction or by allowing a molecule to jump at random to a neighboring subvolume, where diffusion is modeled as a unary reaction with rate proportional to the two dimensional molecular diffusion coefficient divided by the length of a side of the subvolume. In this way, diffusion inside the algorithm becomes another possible event with a propensity function and follows the same update procedure as chemical reaction. Then, the expected time for the next event in a subvolume is calculated similarly to the SSA, including the reaction and diffusion propensities of all molecules contained in that particular subvolume at that particular time. However, time for next events will only be recalculated for those SVs that were involved in the current time step, and they are re-ordered in an event queue.

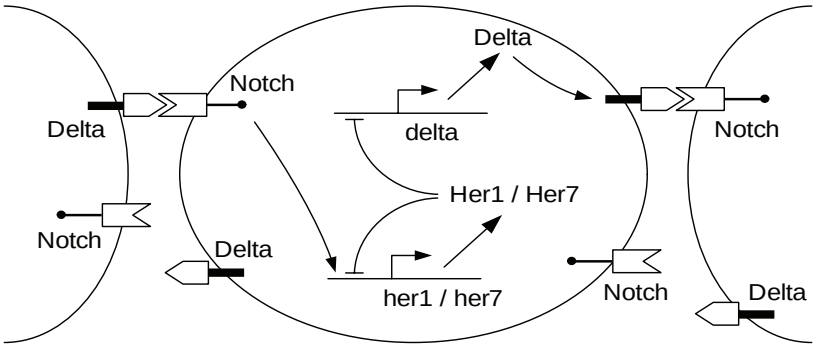
A natural extension of the next subvolume method is to perform τ-leaps that account for one or more diffusion and reaction events, the idea behind Bτ-SSSA (Marquez-Lago and Burrage, 2007). At each iteration, the subvolume with shortest reaction-diffusion τ-leap is selected, which is to be found at the top of the time event queue. Then, all randomly chosen but possible events inside such subvolume are executed, a new τ-leap for all subvolumes that were involved in the current τ-leap is calculated, the time event queue in increasing time is reordered, and the subvolume indicated by the top of the time event queue is chosen. The algorithm is complicated and the reader can refer to the description in the article.

4. CASE STUDIES

In this section we present results from two studies involving Notch signaling molecules. The first model is a model of hes1 auto-inhibition by Hes1 proteins in mouse (Monk, 2003; Barrio et al. 2006). The second model (Figure 5) describes the Delta-Notch dependent synchronization of Her1 and Her7 protein levels in a 1-dimensional array of cells in zebrafish (Lewis, 2003; Horikawa et al., 2006). In this model, the two linked genes her1 and her7 are autorepressed by their own gene products and positively regulated by Delta-Notch cell-cell signaling that leads to oscillatory gene expression in the cells of the presomitic mesoderm (PSM), a region at the tail end of the vertebrate embryo, thus generating regular patterns of somites (embryonic organs that develop into vertebrae and other mammalian repetitive structures (Gonzales and Kageyema, 2007)).

In mammals there are four known Notch genes that encode transmembrane receptors for mediating short-range signaling events. The five known ligands of Notch (Jagged-1,-2 and Delta like-1, -3, and -4) are also transmembrane proteins. At the cell surface, a Notch receptor can interact with one of its ligands in a neighboring cell leading to the release of the Notch intracellular domain (NICD). The subsequent nuclear translocation of NICD results in transcriptional activation of specific genes (Hes and Her/Hesr families) whose corresponding proteins in turn act as transcriptional repressors. There is evidence that

Figure 5. Delta-Notch signaling pathway and the autoinhibition of Notch target genes her1 and her7. Delta proteins in the neighboring cells activate the Notch signal within the cell.



endogeneous NICD acts at very low concentration (Fiúza and Arias, 2007), strongly suggesting a stochastic simulations approach for modeling Delta-Notch signaling. In both models, the transcriptional and translational delays are responsible for the oscillatory behavior. The involved genetic regulation is modeled by delayed Hill type reactions.

4.1 Delta-Notch Signaling: Hes1 and Her1/7

4.1.1 Hes1

The hes1 gene is one of the best characterized genes in the segmentation clocks. Hirata et al. (2004) measured the production of hes1 mRNA (M) and Hes1 protein (P) in mouse. Serum treatments on cultured cells, that have already been shown to induce circadian oscillation by Balsalobre et al. (1998), result in oscillations in expression levels for hes1 mRNA and Hes1 protein in a two hour cycle. Between the oscillatory profiles of mRNA and protein is a phase lag of approximately 15 min. The oscillations in expression continue for 6–12 h and are not dependent on the stimulus but can be induced by exposure to cells expressing Delta. It has been argued that the lag between protein and mRNA oscillation levels of 15 min reflects the time needed for protein degradation. Specifically, the data presented in the paper by Hirata et al. (Figure 1 in Hirata et al., 2004) indicates sustained oscillation of hes1 mRNA over six periods and that oscillation of Hes1 protein that dies away after 6–8 h.

Hirata et al. examined the underlying mechanisms for the observed oscillations and showed that in the presence of the proteasome inhibitor MG132, hes1 mRNA is initially induced but after 3 h it is suppressed because of constant repression of transcription by persistently high protein levels (negative autoregulation). Treatment with cycloheximide leads to sustained increase of hes1 mRNA and blocks its oscillation. A similar effect occurs with overexpression of dnHes1, a dominant-negative form of Hes1 that is known to suppress Hes1 protein activity (Ström et al., 1997). These results reveal that both Hes1 protein synthesis and degradation are needed for oscillations in the expression levels of hes1 mRNA. Other experiments showed that the same mechanisms hold for hes1 mRNA expression levels in the PSM in mouse. Hirata et al. also estimate the half-lives of hes1 mRNA and Hes1 protein to be 24.1 +/- 1.7 min, 22.3 +/- 3.1 min, respectively. Experiments with various protease inhibitors suggest that Hes1 protein is specifically degraded by the ubiquitin–proteasome pathway.

Since the simple negative feedback loop of hes1 mRNA and Hes1 was unable to generate sustained oscillations when modeled as a system of two ODEs, Hirata et al. postulated a Hes1 interacting factor as a third molecular species. Subsequently, they obtained a system of three ODEs that was then able to generate sustained oscillatory behavior. However, there is no direct experimental evidence for such an interacting factor.

Later, it was shown that simple coupled delay differential equations (DDEs), representing the time delays due to transcription and translation, are able to explain the sustained oscillations without recourse to the addition of a third variable (Monk, 2003; Jensen et al., 2003; Lewis, 2000; Bernard et al., 2006). Monk and Jensen et al. proposed the DDE

$$\begin{aligned} \frac{dM}{dt} &= \alpha_M f\left(P\left(t-\tau\right)\right) - \mu_M M \\ \frac{dP}{dt} &= \alpha_P M\left(t\right) - \mu_P P \end{aligned}$$

for the two species, hes1 mRNA (M) and Hes1 (P) and a regulatory Hill function

$$f(P(t)) = \frac{1}{1+(P(t-\tau)/P_0)^{\mathrm{h}}}$$

representing the repression of mRNA production by the binding of Hes1 dimers to the promoter region, with combined transcriptional and translational delay τ, Hill coefficient h and DNA dissociation constant P_0. The reaction rates μ_M and μ_P are the degradation rates of hes1 mRNA and Hes1, respectively, α_M is the maximal mRNA transcription rate in the absence of protein repression, and α_P is the translation rate. See Table II for parameters.

Jensen et al. showed via simulations that for the case $h = 2$, oscillations are only sustained for $\tau > 80$ and there are no oscillations for $\tau < 10$. For $\tau \in (10, 80)$, the period of the damped oscillations is approximately 170 min, which is much greater than the observed period of 120 min. Bernard et al. had shown previously for a modification of the DDE model by Monk that for the experimentally observed period of T=120 min, sustained oscillations can only be obtained for $h \geq 4.1$, $\tau \geq 19.7$. On the other hand, it was argued that since the transcription factor is a Hes1 dimer and there are at least three separate binding sites for Hes1 dimers in the regulatory region of the hes1 gene, an appropriate value of h is at least 2. However, whether h should be as large as 4.1 is debatable.

Barrio et al. (2006) studied the Hes1 negative feedback loop as a discrete, stochastic delay model based on the DDE model by Monk (2003). The chemical kinetics is described by the following reactions:

$$R_1 \colon R_1 \; M \xrightarrow{\mu_M} \emptyset \colon M \xrightarrow{\mu_M} \emptyset \,,$$

$$R_2 \colon R_2 \; P \xrightarrow{\mu_P} \emptyset \colon P \xrightarrow{\mu_P} \emptyset \,,$$

$$R_3 \colon R_3 \; M \xrightarrow{\alpha_P} M + P \colon M \xrightarrow{\alpha_P} M + P \,,$$

$$R_4 \colon R_4 \; P \xrightarrow{\alpha_M, f, \tau} M + P \colon P \xrightarrow{\alpha_M, f, \tau} M + P \,.$$

Table 2. Parameters used in the Hes1-model.

parameter	description	value	Reference
μ_M	Hes1 mRNA degradation rate	0.029 [min^{-1}]	Hirata et al. (2002)
μ_P	Hes1 degradation rate	0.031 [min^{-1}]	Hirata et al. (2002)
α_P	translation rate	1 [min^{-1}]	Monk (2003)
α_M	max. transcription rate	1 [min^{-1}]	Normalized; Monk (2003)
P_0	critical no. of Hes1 protein (Hill function parameter)	10-100	Lewis (2003), Monk (2003)
h	Hill cooperativity factor (Hill function parameter)	2-4	Lewis (2003), Monk (2003)
τ	total delay (transcription, translation, translocation)	10-40 [min]	Monk (2003)

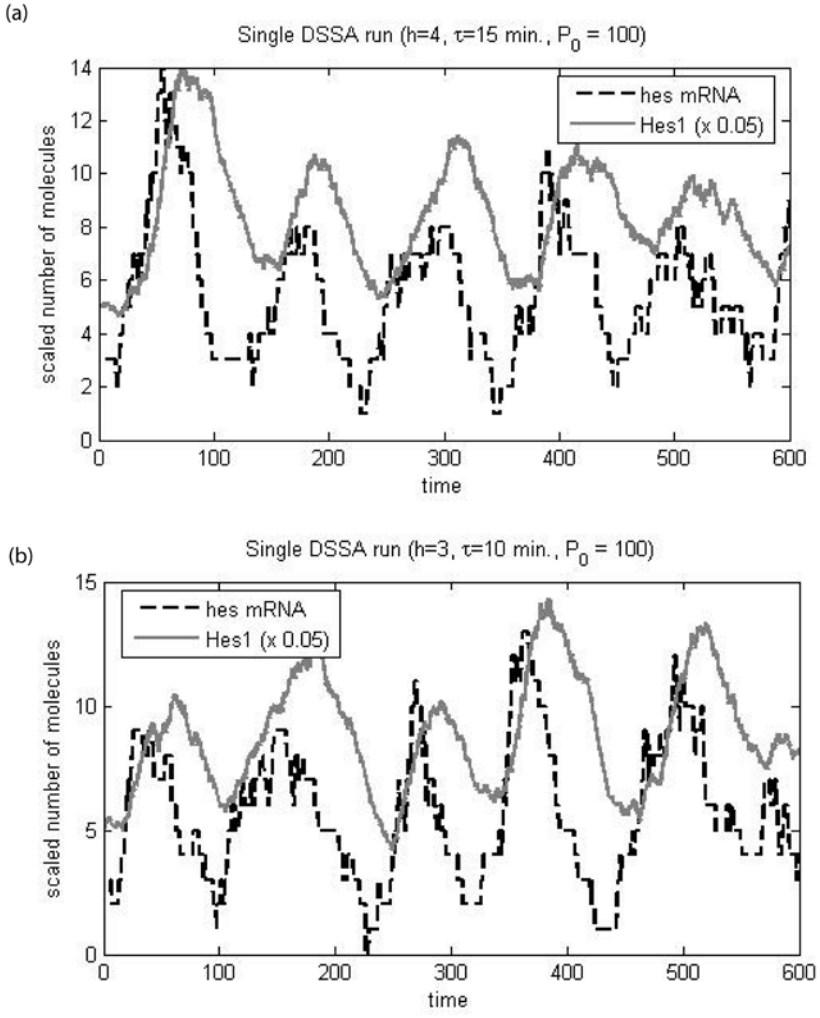
Reactions R_1 and R_2 are the degradations of M and P, respectively. R_3 represents the translation of M and R_4 is the regulated transcription with Hill function f.

By performing discrete stochastic simulations of the model with varying values for h, τ, and P_0 using the DSSA algorithm, Barrio et al. showed that h need not be as large as 4.1 to obtain sustained oscillations when discrete models are used. The results indicate that in the presence of intrinsic noise the critical value of the Hill coefficient, under which the system dynamics does not show sustained oscillations, decreases to just less than 3. Reasonably well-defined sustained regular oscillations could be observed for values of $\tau = 15$ with $h = 4$, and $\tau = 10$ with $h = 3$ (Figure 6). Values for τ lower than 10 result in noisy and irregular delay. By knowing more accurate values for the transcriptional and translational delays an even more accurate prediction of h might be possible and vice-versa.

Barrio et al. (2006) computed the arithmetic mean over 1,000 independent stochastic simulation runs for constant and variable delay. In spite of the differences between individual simulations due to inherent stochasticity, the arithmetic mean showed damped oscillation. This matched the biological experiments where Western-blots of Hes1 from the whole cell population showed damped oscillations that are arrested after eight hours. However, the difference between individual stochastic simulations and the mean suggests that the damping, observed at the whole population level, arises from desynchronization of Hes1 oscillation in individual cells. This was supported by real-time imaging experiments showing that the oscillations in individual cells continue for longer than 8 hours (Masamizu et al., 2006).

The study of the Hes1 negative feedback loop demonstrated the usefulness of the DSSA for chemical kinetics involving delays. Because this approach is very general, it is able to provide deep insights into the relationship between delayed processes, intrinsic noise, and small numbers of molecules in many biological systems.

Figure 6. Single DSSA trajectories for values of (a) $\tau = 15$ *min with* $h = 4$ *, and (b)* $\tau = 10$ *min with* $h = 3$ *(*$P_0 = 100$*).*



4.1.2 Her1/7

The Notch signaling pathway, which includes several signaling molecules (such as Hes1 and Her1/ Her7) in mouse and zebrafish, respectively, plays a key role in the segmentation clock of vertebrates. In (wildlife) zebrafish, about 30–32 somites are formed at a rate of one every 30 min (at 28±C). Although it is suggested that some anterior somites (12) are derived due to some form of dorsal convergence, most somites emerge sequentially from the PSM. It is distinguished between the posterior and anterior parts of the PSM. In zebrafish embryos at a developmental stage of 10 somites, the posterior PSM extends over 25 cells in anterior to posterior axis, which are the precursors for approximately five somites, each about five cells in length. The anterior PSM contains the cells that lead to the next two to three somites.

In zebrafish, the genes her1 and her7 are autorepressed by their own gene products (Her1 and Her7) and positively regulated by Notch signaling (Lewis 2003; Giudicelli and Lewis, 2004) - Figure 5. In both

cases, transcriptional and translational delays are responsible for the oscillatory behavior and determine its period. Additional information on the somite segmentation clock in zebrafish is in Holley (2007) and Lewis and Ozbudak (2007).

Horikawa et al. (2006) performed experiments in which they investigated the system level properties of the segmentation clock in zebrafish. Their main conclusion is that the segmentation clock behaves as a coupled oscillator. The key element is the Notch-dependent intercellular communication, which is regulated by the internal hairy oscillator and whose coupling of neighboring cells synchronizes the oscillations. In one particular experiment, they replaced coupled cells by cells that were out of phase with the remaining cells and showed that at a later stage they still became fully synchronized. Clearly, the intercellular coupling plays a crucial role in minimizing the effects of noise to maintain coherent oscillations.

The stochastic model is based on the chemical reaction models by both Lewis (2003) and Horikawa et al. (2006). Lewis models a single cell and two coupled cells. His work is generalized by Horikawa et al. to a one-dimensional array of *n* cells. For each cell we simulate the dynamics of 6 different species controlled by 12 reactions. Denote by M_{h1_i}, M_{h7_i}, M_{d_i}, P_{h1_i}, P_{h7_i}, and P_{d_i} the species Her1 mRNA, Her7 mRNA, DeltaC mRNA, Her1 protein, Her7 protein and DeltaC protein in a particular cell *i*. For each of the species $S = M_{h1_i}, M_{h7_i}, M_{d_i}, P_{h1_i}, P_{h7_i}, P_{d_i}$, the model contains a degradation reaction

$$S \xrightarrow{c} \emptyset$$

with associated rate constant $c = c_{h1}, c_{h7}, c_d, b_{h1}, b_{h7}, b_d$. The three different proteins P_{h1_i}, P_{h7_i}, P_{d_i} are synthesized with translational delays τ_{h1p}, τ_{h7p}, and τ_{dp}, respectively. The corresponding reactions are

$$S_1 \xrightarrow{c} S_1 + S_2$$

with $(S_1, S_2) = (M_{h1_i}, P_{h1_i})$ or (M_{h7_i}, P_{h7_i}) or (M_{d_i}, P_{d_i}) and associated reaction rate constants $c = a_{h1}, a_{h7}, a_d$. The transcription of M_{h1_i}, M_{h7_i} and M_{d_i} are regulated reactions with transcriptional delays τ_{h1m}, τ_{h7m}, and τ_{dm}, respectively. The reactions are

$$S_1 \xrightarrow{cf} S_1 + S_2$$

with $(S_1, S_2) = (P_{h1_i}, M_{h1_i})$ or (P_{h7_i}, M_{h7_i}) and associated reaction rate constants $c = k_{h1}, k_{h7}$

$$S_1 \xrightarrow{cg} S_1 + S_2$$

with $(S_1, S_2) = (P_{d_i}, M_{d_i})$ and $c = k_d$. As described in detail in Horikawa et al. (2006), the individual negative and positive regulations are modeled using specific Hill functions f and g. For cells *i* with

Table 3. Parameters for the multicellular Her1-Her7 model. Parameter values are taken from Horikawa et al. (2006)

parameter	description	value
b_{h1}, b_{h7}, b_d	Her1/Her7/DeltaC protein degradation rate	0.23 [min^{-1}]
c_{h1}, c_{h7}, c_d	Her1/Her7/DeltaC mRNA degradation rate	0.23 [min^{-1}]
a_{h1}, a_{h7}, a_d	Her1/Her7/DeltaC protein synthesis rate (max.)	4.5 [min^{-1}]
k_{h1}, k_{h7}, k_d	Her1/Her7/DeltaC mRNA synthesis rate (max.)	33 [min^{-1}]
P_0	critical no. of Her1+Her7 protein/cell	40
D_0	critical no. of Delta protein/cell	1000
$\tau_{h1m}, \tau_{h7m}, \tau_{dm}$	time to produce a single Her1/Her7/DeltaC mRNA molecule	12.0, 7.1, 16.0 [min]
$\tau_{h1p}, \tau_{h7p}, \tau_{dp}$	time to produce a single Her1/Her7/ DeltaC protein	2.8, 1.7, 20.5 [min]

$1 < i < n$ (all except for the first and last in the one-dimensional cell array) the Hill function f is defined by

$$f\left(P_{h1_i}, P_{h7_i}, P_{d_{i-1}}, P_{d_{i+1}}\right) = r_h \frac{1}{1 + P_{h1_i} P_{h7_i} \Big/ P_0^2} + r_{hd} \frac{1}{1 + P_{h1_i} P_{h7_i} \Big/ P_0^2} \frac{P_{d_{i-1}} + P_{d_{i+1}}}{2D_0 + P_{d_{i-1}} + P_{d_{i+1}}},$$

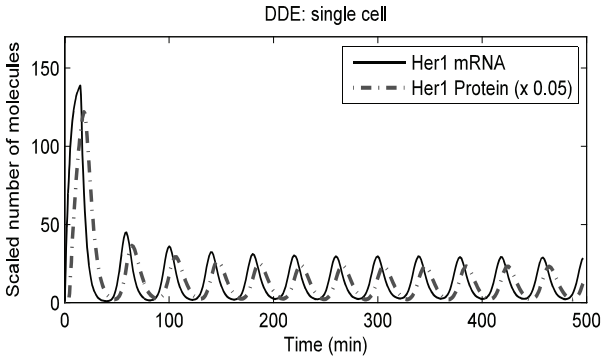
and for cell 1 and *n* it is given by

$$f\left(P_{h1_1}, P_{h7_1}, P_{d_1}\right) = \frac{1}{1 + P_{h1_i} P_{h7_i} \Big/ P_0^2} \frac{P_{d_1} \Big/ D_0}{1 + P_{d_1} \Big/ D_0}$$

$$f\left(P_{h1_n}, P_{h7_n}\right) = \frac{1}{1 + P_{h1_n} P_{h7_n} \Big/ P_0^2} \frac{1}{1 + D_0 \Big/ 500},$$

respectively. The parameters r_h and r_{hd} are weight parameters that determine the balance of internal and external contribution of oscillating molecules. Here, we assume 100% coupling, i.e. $r_{hd} = 1$. For all cells, the Hill function g that describes the inhibition of DeltaC mRNA synthesis by Her1 and Her7 is given by

Figure 7. DDE solution for the Her1/Her7 single cell model



$$g\left(P_{h1_i}, P_{h7_i}\right) = \frac{1}{1 + P_{h1_i} P_{h7_i} / P_0^2} \,.$$

The single cell, single-gene model consists only of 2 species (her1 mRNA and Her1 protein) and 4 reactions. The two degradation and the single translation reactions correspond to those in the *n*-cell model. For the inhibitory regulation of transcription a Hill function with Hill coefficient 2 is assumed (P_{h1} acts as a dimer). The Hill function takes the form

$$f\left(P_{h1}\right) = \frac{1}{1 + P_{h1} / P_0^2} \,.$$

See Table 3 for the full list of model parameters.

A comparison of the DDE solutions with stochastic simulation results of the DSSA and Bτ-DSSA in Leier et al. (2007) and Burrage et al. (2007) revealed differences in the system dynamics. For a single cell, after an initial overshoot, the DDE solution shows completely regular amplitudes and an oscillatory period of approximately 40 minutes (Figure 7). In the intrinsic noise case there are still sustained oscillations but there is some irregularity in the profiles and the oscillatory period is closer to 50 minutes. The time lag (5-7 min) between protein and mRNA is about the same in both cases (Figure 8).

Figure 8. DSSA run for the Her1/Her7 single cell model

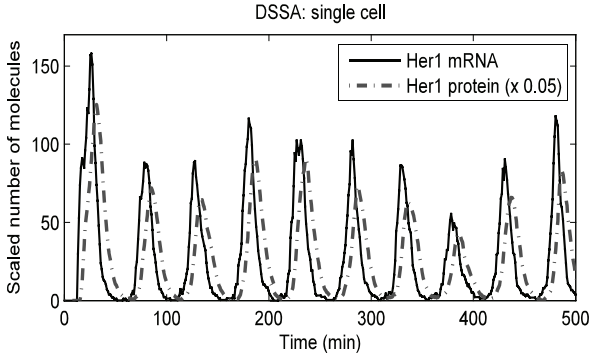
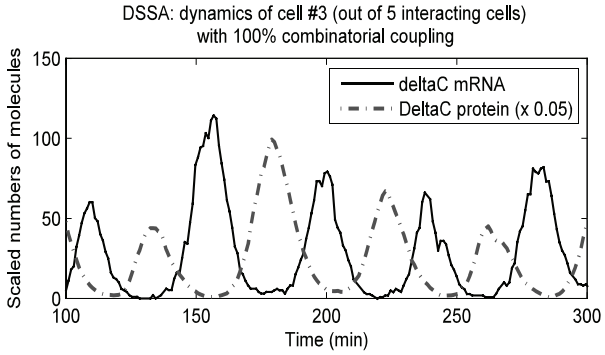


Figure 9. DSSA simulation of five Delta-Notch coupled cells, showing the dynamics of deltaC mRNA and protein in cell three



DSSA simulations of a one-dimensional array of 5 cells exhibit a period of oscillation that is closer to 45 minutes (Figure 9-10). The lag between protein and mRNA is about 25 minutes for DeltaC and about 7 minutes for Her1. Obviously, the cell coupling has some effect on the period of oscillation.

Leier et al. mimic an experiment by Horikawa et al. In both the DDE and the DSSA setting cell 3 (out of 5) is disturbed after a certain time period: after 500 minutes in the DSSA case and 260 minutes in the DDE case, at times when the delta mRNA levels are near their maximum. This is done by resetting all the values for cell 3 to zero at this point. This is meant to represent the experiment of Horikawa et al. in which some of the cells are replaced by oscillating cells that are out of phase. Horikawa et al. observed that nearly all the cells become resynchronized after three oscillations (90 min).

In the DDE setting it takes about 60 minutes for the onset of resynchronization while in the DSSA setting it takes about 180 minutes (Figure 11). The difference can be partly due to the larger number of cells that are experimentally transplanted as well as differences in the cell arrangement between the three-dimensional *in vivo* experiments and the simulated one-dimensional cell array.

This study, although in an early stage, is another example indicating the relevance of both intrinsic noise delay models and continuous deterministic delay models for genetic regulatory systems. Despite some similarities between the dynamics of both the deterministic and stochastic models, the intrinsic

Figure 10. DSSA simulation of five Delta-Notch coupled cells, showing the dynamics of Her1 mRNA and protein in cell three

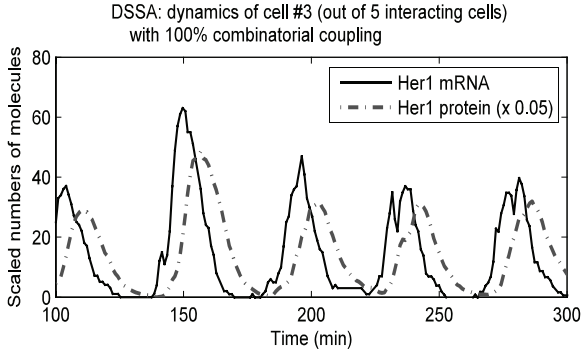
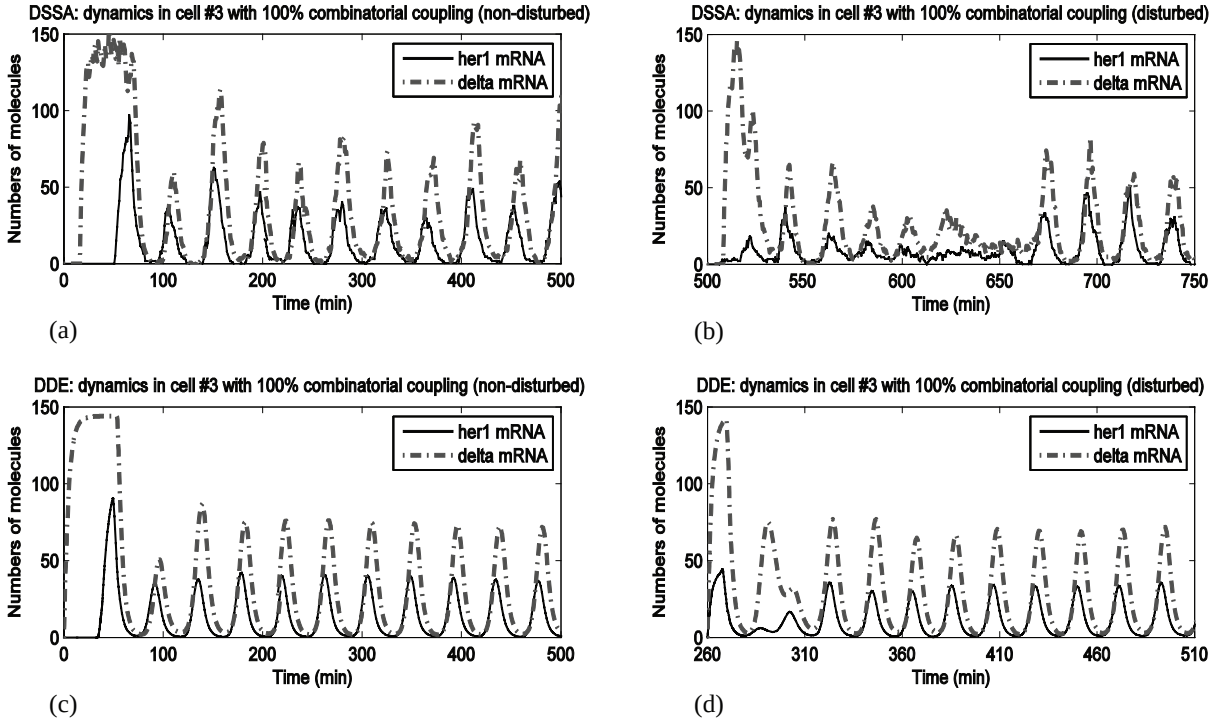


Figure 11. DSSA simulation result and DDE solution for the 5-cell array in the non-disturbed and disturbed setting. The graphs show the dynamics of deltaC and her1 mRNA in cell three. (a,c) DSSA and DDE results in the non-disturbed setting, respectively. (b,d) DSSA and DDE results in the disturbed setting. Initial conditions for cell 3 are set to zero. All other initial molecular numbers stem from the non-disturbed DSSA and DDE results in (a,c) after 500 and 260 minutes, respectively



noise simulations do make some predictions that are different from the deterministic model and that could be verified experimentally.

The reason for limiting the stochastic model to 5 cells is due to the long runtime of individual simulations when using the DSSA. To overcome the issue of small step-sizes, Leier et al. (2008(a)) introduced the Bτ-DSSA (see Section 3.4.2). The significant speed-up (while performing equally accurate as normal DSSA) allows the role of intrinsic noise and delay to be studied for large cellular systems and long time frames. There are many other issues that must be addressed when modeling both delays and intrinsic noise, one of which is how we represent delays. Clearly if delays are to represent complex processes such as transcription and translation, the delays should not be fixed but distributed. Appropriate distributions from which to sample the delays include uniform or truncated normal over some appropriate interval that represents lower and upper bounds for the delays. Other issues include whether it is appropriate to lump delays together into a single delay and how spatial effects associated with, for example, diffusion can be captured in purely temporal models by the use of delays.

5. CONCLUSIONS AND FUTURE DIRECTIONS

In cell biology, cell signaling pathway problems are often tackled with a mix of deterministic temporal models, well mixed stochastic simulators, and/or hybrid methods. But, in fact, three dimensional stochastic spatial modeling of reactions happening inside the cell is sometimes needed in order to fully understand these cell signaling pathways. This is because noise effects, low molecular concentrations, and spatial heterogeneity can all affect the cellular dynamics. However, there are ways in which important effects can be accounted without going to the extent of using these highly resolved spatial simulators. This reduces the overall computation time significantly, while at the same time still being able to capture the essential dynamics.

In this Chapter we have focused on how we can model both intrinsic noise and delayed reactions in a genetic regulatory setting via generalizations of the Stochastic Simulation Algorithm (the DSSA). We have also shown how we can coarsen in both time and space and demonstrated that this can improve the computational performance by several orders of magnitude over the DSSA. We have also shown, through two important applications, why we need algorithms that mimic both noise and delay effects as these approaches can capture the individual cell variability. We have also discussed what form the delays should take: fixed, variable, distributed, etc.

In the delay setting at least, codes based on the algorithms described here are still in their infancy and there is a need to standardize implementations and make these codes available to researchers. Future research must surely focus on multi-scale simulations and there is a great need to develop efficient algorithms that link different temporal and spatial scales – such as genetic regulatory models with those for cellular and organ function. This scientific field is wide open and can promise the dedicated researcher fascinating and rewarding endeavors.

REFERENCES

an der Heiden, U. (1979). Delays in physiological systems. *Journal of Mathematical Biology*, *8*, 345–364.

Balsalobre, A., Damiola, F., & Schibler, U. (1998). A serum shock induces circadian gene expression in mammalian tissue culture cells. *Cell*, *93*, 929–937. doi:10.1016/S0092-8674(00)81199-X

Barrio, M., Burrage, K., & Leier, A. (2006). Oscillatory regulation of Hes1: Discrete stochastic delay modelling and simulation. *PLoS Comp. Bio.*, *2*(9), e117.

Bernard, S., Čajavec, B., & Pujo-Menjouet, L. (2006). Modeling transcriptional feedback loops: The role of Gro/LTE1 in hes1 oscillations. *Phil. Transact. A Math. Phys. Engineering and Science*, *364*, 1155–1170.

Bratsun, D., Volfson, D., & Tsimring, L. S. (2005). Delay-induced stochastic oscillations in gene regulation. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 14593–14598. doi:10.1073/pnas.0503858102

Burrage, K., Burrage, P. M., Leier, A., et al. (2008). Stochastic delay models for molecular clocks and somite formation. In *Proceedings of SPIE*, 68020Z.

Burrage, K., Hegland, M., MacNamara, S., et al. (2006). A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. In A. N. Langville & W. J. Stewart (Eds.), *Proceedings of the Markov 150th Anniversary Conference* (pp. 21-38). Boson Books.

Cai, X. (2007). Exact stochastic simulation of coupled chemical reactions with delays. *The Journal of Chemical Physics*, *126*(12), 124108–124116. doi:10.1063/1.2710253

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2005). Avoiding negative populations in explicit tau leaping. *The Journal of Chemical Physics*, *123*, 054104. doi:10.1063/1.1992473

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2006). Efficient stepsize selection for the tau-leaping method. *The Journal of Chemical Physics*, *124*, 044109. doi:10.1063/1.2159468

Chatterjee, A., & Vlachos, D. G. (2006). Multiscale spatial Monte Carlo simulations: Multigriding, computational singular perturbation, and hierarchical stochastic closures. *The Journal of Chemical Physics*, *124*, 064110. doi:10.1063/1.2166380

Chatterjee, A., & Vlachos, D. G. (2006). Temporal acceleration of spatially distributed kinetic Monte Carlo simulations. *Journal of Computational Physics*, *211*(2), 596–615. doi:10.1016/j.jcp.2005.06.004

Chatterjee, A., Vlachos, D. G., & Katsoulakis, M. A. (2005). Binomial distribution based τ-leap accelerated stochastic simulation. *The Journal of Chemical Physics*, *124*, 044109.

Elf, J., Doncic, A., & Ehrenberg, M. (2003). Mesoscopic reaction-diffusion in intracellular signaling. *Proceedings of the Society for Photo-Instrumentation Engineers*, *5110*, 114–124. doi:10.1117/12.497009

Elf, J., & Ehrenberg, M. (2004). Spontaneous separation of bistable biochemical systems into spatial domains of opposite phases. *Systems Biology*, *2*, 230. doi:10.1049/sb:20045021

Elowitz, M. B., & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, *403*, 335–338. doi:10.1038/35002125

Fiúza, U.-M., & Arias, A. M. (2007). Cell and molecular biology of Notch. *The Journal of Endocrinology*, *194*, 459–474. doi:10.1677/JOE-07-0242

Gibson, M. A., & Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, *104*, 1876–1889. doi:10.1021/jp993732q

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, *81*, 2340–2361. doi:10.1021/j100540a008

Gillespie, D. T. (2000). The chemical Langevin equation. *The Journal of Chemical Physics*, *113*, 297–306. doi:10.1063/1.481811

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, *115*(4), 1716–1733. doi:10.1063/1.1378322

Gillespie, D. T., & Petzold, L. R. (2003). Improved leap-size selection for accelerated stochastic simulation. *The Journal of Chemical Physics*, *119*(16), 8229–8234. doi:10.1063/1.1613254

Giudicelli, F., & Lewis, J. (2004). The vertebrate segmentation clock. *Current Opinion in Genetics & Development*, *14*(4), 407–414. doi:10.1016/j.gde.2004.06.014

Gonzales, A., & Kageyema, R. (2007). Practical lessons from theoretical models about the somitogenesis. *Gene Regulation and Systems Biology*, *1*, 35–42.

Goodwin, B. C. (1965). Oscillatory behavior in enzymatic control processes. *Advances in Enzyme Regulation*, *3*, 425–438. doi:10.1016/0065-2571(65)90067-1

Hattne, J., Fange, D., & Elf, J. (2005). Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics (Oxford, England)*, *21*, 2923. doi:10.1093/bioinformatics/bti431

Hirata, H., Yoshiura, S., & Ohtsuka, T. (2002). Oscillatory expression of the bHLH factor Hes1 regulated by a negative feedback loop. *Science*, *298*, 840–843. doi:10.1126/science.1074560

Holley, S. A. (2007). The genetics and embryology of zebrafish metamerism. *Developmental Dynamics*, *236*(6), 1422–1449. doi:10.1002/dvdy.21162

Horikawa, K., Ishimatsu, K., & Yoshimoto, E. (2006). Noise-resistant and synchronized oscillation of the segmentation clock. *Nature*, *441*(7094), 719–723. doi:10.1038/nature04861

Jensen, M. H., Sneppen, K., & Tiana, G. (2003). Sustained oscillations and time delays in gene expression of protein Hes1. *FEBS Letters*, *541*, 176–177. doi:10.1016/S0014-5793(03)00279-5

Kurtz, T. G. (1972). The relationship between stochastic and deterministic models of chemical reactions. *The Journal of Chemical Physics*, *57*, 2976–2978. doi:10.1063/1.1678692

Leier, A., Márquez-Lago, T., & Burrage, K. (2008). Modeling intrinsic noise and delays in chemical kinetics of coupled autoregulated oscillating cells. *Int. J. Multiscale Computational Engineering, 6*(1).

Leier, A., Márquez-Lago, T., & Burrage, K. (2008). Generalized binomial τ-leap method for biochemical kinetics incorporating both delay and intrinsic noise. *The Journal of Chemical Physics*, *128*, 205107. doi:10.1063/1.2919124

Lewis, J. (2003). Autoinhibition with transcriptional delay: A simple mechanism for the zebrafish somitogenesis oscillator. *Current Biology*, *13*, 1398–1408. doi:10.1016/S0960-9822(03)00534-7

Lewis, J., & Ozbudak, E. M. (2007). Deciphering the somite segmentation clock: Beyond mutants and morphants. *Developmental Dynamics*, *236*(6), 1410–1415. doi:10.1002/dvdy.21154

MacNamara, S., Burrage, K., & Sidje, R. (2008). Multiscale modeling of chemical kinetics via the master equation. *SIAM J. Multiscale Modelling and Simulation Multiscale Modeling & Simulation, 6*(4).

Marquez-Lago, T., & Burrage, K. (2007). Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics. *The Journal of Chemical Physics*, *127*(10), 104101. doi:10.1063/1.2771548

Masamizu, Y., Ohtsuka, T., & Takashima, Y. (2006). Real-time imaging of the somite segmentation clock: Revelation of unstable oscillators in the individual presomitic mesoderm cells. *Proceedings of the National Academy of Sciences of the United States of America, 103*(5), 1313–1318. doi:10.1073/pnas.0508658103

Monk, N. (2003). Oscillatory expression of Hes1, p53, and NF-κB driven by transcriptional time delays. *Current Biology, 13,* 1409–1413. doi:10.1016/S0960-9822(03)00494-9

Morton-Firth, C. J., & Bray, D. (1998). Predicting temporal fluctuations in an intracellular signalling pathway. *Journal of Theoretical Biology, 192,* 117–128. doi:10.1006/jtbi.1997.0651

Munsky, B., & Khammash, M. (2006). The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics, 124,* 044104. doi:10.1063/1.2145882

Nicolau, D. V. Jr, Burrage, K., & Parton, R. G. (2006). Identifying optimal lipid raft characteristics required to promote nanoscale protein-protein interactions on the plasma membrane. *Molecular and Cellular Biology, 26,* 313–323. doi:10.1128/MCB.26.1.313-323.2006

Peng, X., Zhou, W., & Wang, Y. (2007). Efficient binomial leap method for simulating chemical kinetics. *The Journal of Chemical Physics, 126,* 224109. doi:10.1063/1.2741252

Rathinam, M., Petzold, L. R., & Cao, Y. (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics, 119,* 12784. doi:10.1063/1.1627296

Resat, V. H., Wiley, H. S., & Dixon, D. A. (2001). Probability-weighted dynamic Monte Carlo method for reaction kinetics simulations. *The Journal of Physical Chemistry B, 105,* 11026–11034. doi:10.1021/jp011404w

Ström, A., Castella, P., & Rockwood, J. (1997). Mediation of NGF signaling by post-translational inhibition of HES-1, a basic helix–loop–helix repressor of neuronal differentiation. *Genes & Development, 11,* 3168–3181. doi:10.1101/gad.11.23.3168

Tian, T., & Burrage, K. (2004). Binomial leap methods for simulating stochastic chemical kinetics. *The Journal of Chemical Physics, 121*(21), 10356–10364. doi:10.1063/1.1810475

Tian, T., Burrage, K., Burrage, P. M., & Carletti, M. (2007). Stochastic delay differential equations for genetic regulatory networks. *Journal of Computational and Applied Mathematics, 205*(2), 696–707. doi:10.1016/j.cam.2006.02.063

Turner, T., Schnell, S., & Burrage, K. (2004). Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry, 28,* 165. doi:10.1016/j.compbiolchem.2004.05.001

ADDITIONAL READING

Anderson, D. F. (2007). A modified next reaction method for simulating chemical systems with time dependent propensities and delays. *The Journal of Chemical Physics, 127,* 214107. doi:10.1063/1.2799998

Anderson, D. F. (2008). Incorporating postleap checks in tau-leaping. *The Journal of Chemical Physics, 128,* 054103. doi:10.1063/1.2819665

Auger, A., Chatelain, P., & Koumoutsakos, P. (2006). R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. *The Journal of Chemical Physics, 125*(8), 084103. doi:10.1063/1.2218339

Bessho, Y., & Kageyama, R. (2003). Oscillations, clocks, and segmentation. *Current Opinion in Genetics & Development, 13,* 379–384. doi:10.1016/S0959-437X(03)00083-2

Burrage, K., Tian, T., & Burrage, P. M. (2004). A multiscaled approach for simulating chemical reaction systems. *Progress in Biophysics and Molecular Biology*, *85*, 217–234. doi:10.1016/j.pbiomolbio.2004.01.014

El Samad, H., Khammash, M., & Gillespie, D. (2002). Stochastic modeling of gene regulatory networks. *Int. J. Robust and Nonlinear Control., 15,* 691–711. doi:10.1002/rnc.1018

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, *58,* 35–55. doi:10.1146/annurev.physchem.58.032806.104637

Goutsias, J. (2007). Classical versus stochastic kinetics modeling of biochemical reaction systems. *Biophysical Journal*, *92*, 2350–2365. doi:10.1529/biophysj.106.093781

Kitano, H. (2000). Computational systems biology. *Nature*, *420,* 206–210. doi:10.1038/nature01254

Márquez Lago, T., Leier, A., & Burrage, K. (in preparation for submission). Modeling molecular translocation processes with a stochastic delay simulation algorithm.

McAdams, H. H., & Arkin, A. (1999). It's a noisy business! Genetic regulation at the nanomolar scale. *Trends in Genetics*, *15*, 65–69. doi:10.1016/S0168-9525(98)01659-X

Puchalka, J., & Kierzek, A. M. (2004). Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophysical Journal*, *86*, 1357–1372. doi:10.1016/S0006-3495(04)74207-1

Raser, J. M., & O'Shea, E. K. (2005). Noise in gene expression: Origins, consequences, and control. *Science*, *309,* 2010–2013. doi:10.1126/science.1105891

Saga, Y., & Takeda, H. (2001). The making of the somite: Molecular events in vertebrate segmentation. *Nature Reviews. Genetics*, *2,* 835–884. doi:10.1038/35098552

Ullah, M., Schmidt, H., & Cho, K. (2006). Deterministic modelling and stochastic simulation of biochemical pathways using MATLAB. *Systems Biology . IEEE Proceedings*, *153*, 53–60.

Zheng, Q., & Ross, J. (2001). Comparison of deterministic and stochastic kinetics for nonlinear systems. *The Journal of Chemical Physics*, *94*, 3644–3648. doi:10.1063/1.459735

Chapter 8
Modeling Gene Regulatory Networks with Delayed Stochastic Dynamics

Andre S. Ribeiro
Tampere University of Technology, Finland

John J. Grefenstette
George Mason University, USA

Stuart A. Kauffman
University of Calgary, Canada

ABSTRACT

We present a recently developed modeling strategy of gene regulatory networks (GRN) that uses the delayed stochastic simulation algorithm to drive its dynamics. First, we present experimental evidence that led us to use this strategy. Next, we describe the stochastic simulation algorithm (SSA), and the delayed SSA, able to simulate time-delayed events. We then present a model of single gene expression. From this, we present the general modeling strategy of GRN. Specific applications of the approach are presented, beginning with the model of single gene expression which mimics a recent experimental measurement of gene expression at single-protein level, to validate our modeling strategy. We also model a toggle switch with realistic noise and delays, used in cells as differentiation pathway switches. We show that its dynamics differs from previous modeling strategies predictions. As a final example, we model the P53-Mdm2 feedback loop, whose malfunction is associated to 50% of cancers, and can induce cells apoptosis. In the end, we briefly discuss some issues in modeling the evolution of GRNs, and outline some directions for further research.

INTRODUCTION

After sequencing the genomes of various organisms, understanding the integrated behavior of gene regulatory networks (GRN), taken in the large sense to comprise genes, RNA, proteins, microRNA and other molecules that mutually interact to control the dynamical behavior of the GRN within and between

DOI: 10.4018/978-1-60566-685-3.ch008

cells, has emerged as a fundamental problem in Systems Biology. So far there is only partial knowledge of the regulatory network structure and "logic" driving GRNs' dynamical behavior. Nevertheless, one can begin to address questions using the known features of these networks, by constructing the family, or ensemble, of all networks consistent with those observations.

This "ensemble" approach (Kauffman, 2004) studies the expected properties of members of the ensemble and predicts new observables to test against the dynamical behavior of cells and tissues. It is a further profound issue whether real networks are generic to any ensemble, given 3 billion years of evolution and natural selection.

There are three frameworks to analyze GRNs. At the most detailed level, one considers the chemical master equation of the detailed behavior of all components in members of some ensemble of networks. This can be done (McAdams et al, 1997) using the stochastic simulation algorithm (SSA) (Gillespie, 1977). Such models are inherently stochastic. Understanding the consequences of such noise is itself a critical problem and is focused in this chapter.

At a second level of abstraction, one considers systems of deterministic nonlinear ODEs (Mestl et al, 1995) capturing, in some sense, the mean field behavior of the real noisy stochastic networks. Since the number of copies of regulatory molecules in real system can be very small (from one to a few), such deterministic equations are, at best, an approximation, and several recent works have shown limitations of this method (Lipshtat et al, 2006). The addition of noise in Langevin equations (Toulouse et al, 2005), e.g., remains to be shown to capture the true nature of cellular dynamical noise.

At a still higher level of abstraction, one can consider models where gene states, time and other components are all discrete. While furthest from the detailed description, such models have the advantages of allowing the study of very large networks, with thousands of model genes. In particular, random Boolean networks (RBN) have been the subject of considerable analysis (Kauffman, 1969).

Here, we present the latest modeling strategy of GRNs (Ribeiro et al, 2006a), which aims to capture the relevant features of GRN to achieve simulations as realistic as possible. The dynamics is driven by the delayed SSA (Roussel & Zhu, 2006), which allows modeling multiple time-delayed reactions while maintaining a realistic account of molecular noise. We show evidence of its validity and accuracy at a detailed level.

The chapter is organized as follows. First, we describe recent experimental measurements that reveal key features that should be reflected in models of gene expression and gene-gene interactions. After that, we describe the SSA and the delayed SSA.

Next, a model for single gene expression is presented. It is shown that this model accurately reproduces recent measurements of gene expression at the single molecule level. Based on this model, a model of GRNs is proposed (Ribeiro et al, 2006a). Importantly, this modeling strategy allows applying the ensemble approach (Kauffman, 2004), which consists in simulating the dynamics of many GRNs with similar features, and extracting general properties of the dynamics from the resulting time series.

Subsequently, examples of applications of the modeling strategy are presented. A model of a toggle switch shows the relevance of including time delays in gene expression. To show the ability of modeling complex chemical pathways, we present a model of the P53-Mdm2 chemical feedback-loop, associated with important biochemical pathways in cells, responsible for responding to external stresses and apoptosis. The final section includes some preliminary studies on the evolution of these models of GRNs.

GENE EXPRESSION AT A DETAILED LEVEL

Two key features in gene expression should be considered in dynamical models of GRNs. First, experimental measurements of gene expression show that the underlying dynamics has non-negligible stochastic fluctuations. This critical feature cannot be ignored since genes exist in small copy numbers in the genome (from one to a few), and some express at very low rates. Second, transcription and translation are multiple-step processes involving a large number of reactions, and thus take a non-negligible time to be complete once initiated. We describe recent experiments that establish the stochasticity in gene expression and measurements of time durations involved in gene expression.

Stochastic Nature of Gene Expression

Stochastic fluctuations of gene expression were proven to have a significant role at the single-cell level (Elowitz et al, 2002), e.g., controlling probabilistic cellular differentiation pathway choice (Arkin et al, 1998).

A study (Arkin et al, 1998) proved the relation between differentiation pathway selection (that is, if a cell type A differentiates to either a cell type B or a cell type C), and the stochastic nature of genes' expression. Fluctuations in gene expression produce erratic time patterns of protein production in individual cells and wide diversity in protein concentrations across cell populations, explaining the observed probabilistic differentiation pathway choice. It has been experimentally observed that cell populations, initially homogeneous, separate into distinct phenotypic sub-populations, due to stochastic fluctuations. Importantly, it was also shown that the regulatory proteins exist in very low cellular concentrations and compete in the control of the pathway switch points of possible differentiation pathways.

A model of gene expression (without time delays), driven by the SSA, was able to match experimentally observed ratios of cells choosing each of the differentiation pathways and also the dynamics at the single cell level (Arkin et al, 1998). The same model also mimicked correctly the production of proteins from an activated promoter, in short bursts of variable numbers of proteins whose occurrence is separated by time intervals of random duration (McAdams & Arkin, 1997).

The fact that only a small number of molecules are involved in these processes, that genes' promoter regions exist in very low copy numbers in the cell (Becskei & Serrano, 2000), and that gene expression is stochastic leads to the conclusion that conventional deterministic kinetics, even with noise terms, cannot predict the statistics of regulatory systems that produce probabilistic outcomes.

Recent experimental measurements confirmed that noise cannot be neglected in GRNs dynamics. Quantitative fluorescence measurements of gene expression products (Süel et al, 2006)(Süel et al, 2007) showed that genes' expression and cells' differentiation are highly noisy, and established that certain types of cellular differentiation are probabilistic and transient. They reported cases of cells' going back and forth from one cell type to another, without external perturbations.

These and other experiments showed how populations of cells, genetically identical and in the same environmental conditions, have individual cells with distinct phenotypes, implying the intrinsic stochasticity of GRNs.

Time Delays in Transcription and Translation

Among the many steps involved in gene expression, some are time consuming, such as transcripts creation and modification, mature mRNA transport to the cytoplasm (in Eukaryotes), mRNA translation (Ota et al, 2003), and post-translation protein modifications and folding. The time intervals between these sub-processes play important roles in biochemical dynamics and must be incorporated in models with a genetic regulatory component (Ribeiro et al, 2006a).

Transcription elongation is the process by which the RNA polymerase (RNAp) slides along the template strand and adds bases to the transcript, according to the DNA sequence. Its duration depends on the gene length and the RNAp transcription speed. Also, the duration varies between different events even for the same gene, because it depends on the rate by which the reactions occur, and these are stochastic events. Measurements of elongation times showed that the velocities of different transcription events followed a normal distribution (Davenport et al, 2000).

Although stochastic models of GRNs, using only non-delayed reactions can explain experimental data regarding gene expression fluctuations (Raser & O'Shea, 2004), these studies focused on steady state dynamics, where delayed and non-delayed models have the same results after an initial transient. Models of more complex GRN (e.g., involving feedback mechanisms), require modeling transcription and translation as time-delayed reactions. Thus, in the model here presented, the time duration of transcription, translation, etc, is included to capture the features of transients.

MULTI-DELAYED STOCHASTIC SIMULATION ALGORITHM

Stochastic Simulation Algorithm

The Stochastic Simulation Algorithm (SSA) (Gillespie, 1977), a Monte Carlo simulation of the chemical master equation, is an exact procedure for numerically simulating the time evolution of a well-stirred reacting system.

Each chemical species quantity is treated as an independent variable and each reaction is executed explicitly. Time evolves in discrete steps, with each step being the execution of a specific reaction at a specific time. After a reaction is executed, the number of molecules of each of the affected species is updated according to the reaction formula, and the algorithm advances to the next event. Because each reaction and the time for the next reaction to occur are independent of the preceding ones, the temporal evolution of the system is a Markov process.

The algorithm is exact in the sense that each simulation of a system of chemical reactions, in the conditions required by the SSA, provides an exact temporal trajectory, matching one of the system's possible trajectories in its state space. The necessary condition for the SSA to be valid for any chemical system is that such system is kept "well-stirred" during the simulation, either by direct stirring or by requiring that non-reactive molecular collisions occur far more frequently than reactive molecular collisions (Gillespie, 1977). For the collision probability of two molecules to be spatially homogenous, one must assume that, each time a reaction occurs due to a collision between two potentially reacting molecules, this event will be followed by many non-reactive collisions, which cause the molecules to be once again uniformly distributed in space before the next reactive event occurs.

Each reaction rate constant, cμ, is dependent on the reactive radii of the molecules involved in the reaction and their average relative velocities. The velocities depend on the temperature of the system and the individual molecular masses. After setting the initial species populations Xi and reactions rate constants cμ, the SSA calculates the propensity aμ = cμ.hμ, for all possible reactions. The variable hμ is the number of distinct molecular reactants combinations available at a given moment in time.

The SSA then generates two random numbers, r1 and r2, which are used to compute τ, the time interval until the next reaction occurs, and μ, which determines which reaction occurs.

Finally, the system time t is increased by τ and the Xi quantities are adjusted to account for the occurrence of reaction μ, assuming that it occurred instantaneously. This process is repeated until no more reactions can occur, or during a user defined time interval.

As seen in the formulation of the algorithm, the probabilities for events to occur are converted into the expected time it takes until they actually occur. That allows computing the system state temporal evolution. The SSA goes as follows (Gillespie, 1977):

Step 0 (Initialization). Input the desired values for the M reaction rate constants c_1,. . ., c_M and the N initial molecular population numbers X_1,. . ., X_N. Set the time variable t and the reaction counter n both to zero. Initialize the unit-interval uniform random number generator (URN).

Step 1. Calculate and store the M quantities, $a_1 = c_1.h_1$,..., $a_M = c_M.h_M$ for the current molecular population numbers, where h_μ, is the number of distinct molecular reactant combinations available, given the system current state (X_1,..., X_N)(μ = 1,. . .,M). Calculate and store as a_0 the sum of the M a_i, values.

Step 2. Generate two random numbers r_1 and r_2 from a unitary uniform distribution, and calculate τ and μ according to: $\tau = (1/a_0).\ln(1/r_1)$, and μ is an integer such that: $\sum_{v=1}^{\mu-1} a_v < r_2.a_0 < \sum_{v=1}^{\mu} a_v$.

Step 3. Using the τ and μ values obtained in step 2, increase t by τ, and adjust the molecular population levels to reflect the occurrence of the reaction chosen to occur. Then increase the reaction counter n by 1 and return to step 1.

Delayed Stochastic Simulation Algorithm

The delayed SSA captures two important features of GRN dynamics: stochastic dynamics and the existence of events whose time duration for completion, once initiated, cannot be ignored. The delayed SSA (Roussel and Zhu, 2006), which is a generalization of the algorithm proposed in (Bratsun et al, 2005), proceeds as follows:

Step 1. Set t = 0, stop time = t_{stop}, read initial number of molecules and reactions, create empty wait list L.

Step 2. Do an SSA step for input events to get the next reaction event R1 and its occurrence time t_1.

Step 3. If $t_1 + t < t_{min}$ (the least time in L), set $t \leftarrow t + t_1$. Update number of molecules by performing R1, adding delayed products into L as necessary.

Step 4. If $t_1 + t \geq t_{min}$, set $t \leftarrow t_{min}$. Update number of molecules by releasing the first element in L.

Step 5. If $t < t_{stop}$, go to step 2.

To simulate time-delayed reactions one needs a "waiting list". Each reaction product that takes time τ* to be created after the reaction occurrence, is, when the reaction occurs, placed in a waiting list, until that time τ* elapses. At that moment, the product is released into the system and becomes available for possible reactions.

Take as an example, the following chemical reaction: $A + B \xrightarrow{k} A + C(\tau^*)$. When this reaction is selected to occur, the number of molecules A is kept constant, a single molecule B is immediately removed from the system, and a molecule C is placed on a waitlist, and will be released into the system τ* seconds after the reaction occurred.

MODEL OF GENE EXPRESSION AS A MULTI-DELAYED REACTION

To model transcription and translation in prokaryote cells, accounting for delays, the following reactions (1) and (2) are used (Zhu et al, 2007):

$$\mathrm{Pro}(t) + RNAp(t) \xrightarrow{k_{transc}} \mathrm{Pro}(t+\tau_1) + \ RBS(t+\tau_1) + RNAp(t+\tau_2) + R(t+\tau_2) \quad (1)$$

$$\mathrm{Ribosome}(t) + RBS(t) \xrightarrow{k_{transl}} \mathrm{Ribosome}(t+\tau_3) + \ RBS(t+\tau_3) + P(t+\tau_4) \quad (2)$$

Reaction 1 models transcription. An RNAp binds to a gene's promoter and transcribes the gene. The promoter remains unavailable for further reactions for τ_1 seconds after the reaction takes place, and the part of the RNA (the ribosome binding site, RBS), to which ribosomes can bind to and translate, is also produced in τ_1 seconds. When transcription is complete, at τ_2, the RNAp and a complete RNA molecule, R, are released for further reactions in the system. Reaction 2 models translation and is also a multiple time delayed reaction, similar to the one for transcription.

In eukaryotes, instead of RBS, complete RNA molecules are the substrate for translation, since the RNA has to leave the nucleus before it can be translated.

In some cases this model can be simplified, modeling transcription and translation in a single step, e.g., if all control mechanisms of gene expression act upon the protein level rather than at the RNA level. In this case (Ribeiro et al, 2006a), when an RNAp binds to a gene promoter, if reacting, the output is, besides the gene promoter region and the RNAp, an active protein (reaction 3):

$$\mathrm{Pro}(t) + RNAp(t) \xrightarrow{k_{transc\&transl}} \mathrm{Pro}(t+\tau_1) + \ RNAp(t+\tau_2) + n.P(t+\tau_3) \quad (3)$$

The protein P_i in (3) is created by RNA translation. The n_i variable is an integer, associated to the rate of translation. This variable can be drawn from a distribution of integers, each time the reaction occurs, and can differ for each gene.

GENE REGULATORY NETWORK MODEL

In real cells, the GRN is involved in most cellular processes. Some of its products are transcription factors and co-factors that regulate the activity of downstream genes. Other products regulate cellular chemistry, which by feedback chemical pathways can regulate several genes' expression.

In our model, regulation of a gene's expression by another gene expression product is assumed to occur via the proteins' expressed by the genes. A protein can act as either as a transcription factor that binds to the operator site of another gene (changing its expression propensity) or it can act as a repressor by, for example, degrading another gene's proteins. Proteins can form homodimers, heterodimers, or higher order polymers that can feedback into the GRN. The set of interactions among genes, via their products of expression, defines the network topology.

Genes are represented by their promoter regions. The promoter region includes the initiation sequence, to which the RNAp can bind and begin transcription, and the operator sites region, to which transcription factors can bind and change the genes' transcription reaction propensity.

Transcription factors can act as activators or inhibitors of gene transcription. Genes can have multiple operator sites, and the effect of multiple transcription factors can be, in general, combinatorial. E.g., a certain protein can act as inhibitor if it is the only transcription factor bound to the gene, but can act as an activator if another specific transcription factor is also bound to the gene in another operator site.

Since genes can have multiple operator sites, the following notation is used: $Pro_{i,(op)}$, such that, i is the gene identification index, and (op), is an array of all operator sites, and its values represent the state of each of the gene's operator sites. Such state consists of having or not transcription factors bound to the operator site and what transcription factor is bound to the site, if any.

For each combination of inputs states (promoter state), a regulating function is assigned that determines the gene's expression rate in such state. E.g., imagine gene 7 has 2 operator sites. Assume that p_1 and p_2 can bind to operator site 1, while p_3 can bind to operator site 2, of gene 7. If, at any given moment in the simulation, p1 is bound to site 1, and no protein is bound to site 2, the operator is in the state $Pro_{i,(p1,\ 0)}$. Another possible state would be $Pro_{i,(p2,\ p3)}$. Depending on its promoter occupancy state, the gene is either repressed or activated at a certain rate. A fraction of genes can be assigned to have basic level of expression (a promoter with no transcription factors bound to it can be transcribed by an RNAp), while others do not have that ability.

This procedure to design GRNs can be seen as a generalization of the procedure used to create RBNs (Kauffman, 1969). In RBNs one assigns random Boolean functions to each gene, creating a combinatorial logic. This can be attained in our model in three ways: [i] allowing reactions between genes' expression products, and then assign such resulting complexes as activator or repressor of another gene; [ii] allowing a gene to have more than one operator site and randomly assigning the effects of all the possible binding combinations as activations or inhibitions; [iii] allowing different genes expression products to bind competitively to a single binding site, each with a different effect on the gene transcription rate.

These features allow all Boolean functions and topologies of interactions to have representations in this model. However, a single RBN could be mapped to an infinite number of different GRNs using our model, since many parameters, such as rate constants and delays, are not defined in RBNs.

This model of GRNs (Ribeiro et al, 2006a) has been implemented by a software package called SGNSim (Ribeiro & Lloyd-Price, 2007). In SGNSim, GRNs are generated from the following reactions (4 to 11).

For gene i = 1,…,N there is basal transcription reaction of promoter Proi by one RNAp (reaction 4). The model includes transcription reactions for promoters with specific sets of transcription factors bound to it (reaction 5) and translation of RNA by ribosomes (Rib) into proteins (reaction 6). These are all time-delayed reactions. The delays are represented by a τ variable and differ between products of each reaction, and between similar reactions for different genes (since these have different lengths).

The binding/unbinding of a transcription factor from operator site j of a gene i, are represented in reactions 7. If the complex is unable to transcribe, these two reactions represent repression/unrepression else, are activation/inactivation reactions. Note that reaction 7 is bidirectional, corresponding to the binding of the repressor, and its spontaneous unbinding.

Reaction 9 represents the loss of repression due to an external "re-activator" protein that removes the repressor from the operator site.

Decay of RNA, represented by its ribosome binding site RBS, and proteins, occur via the reactions 10. Decay of a protein while bound to a promoter occurs via reaction 8. Finally, proteins polymerization (here, limited to dimers for simplicity) and the inverse reaction, occur via the bidirectional reactions 11. Unless time delays are explicitly represented in the products of the reactions (here represented using the notation X(τ)), all events, including depletion of reactants and appearance of products, occur instantaneously at the time the reaction takes place, t:

$$\mathrm{Pro_i} + RNAp \xrightarrow{k_{transc}} \mathrm{Pro_i}(\tau_1) + \ RBS_\mathrm{i}(\tau_1) + RNAp(\tau_2) + R_\mathrm{i}(\tau_2) \tag{4}$$

$$\mathrm{Pro}_{i,\overline{\{op\}}} + RNAp \xrightarrow{k_{t1,op1}} \mathrm{Pro}_{i,\overline{\{op\}}}(\tau_i^1) + \ RBS_i(\tau_i^1) + RNAp(\tau_i^2) + R_\mathrm{i}(\tau_2) \tag{5}$$

$$\mathrm{Rib} + RBS_i \xrightarrow{k_{tr1,RBS}} \mathrm{Rib}(\tau_3) + \ RBS_i(\tau_3) + p_i(\tau_4) \tag{6}$$

$$\mathrm{Pro}_{i,(\ldots,j,\ldots)} + p_w \underset{\mathrm{k_{w,ij}}}{\overset{\mathrm{k_{ij,w}}}{\rightleftharpoons}} \mathrm{Pro}_{i,(\ldots,p_w,\ldots)} \tag{7}$$

$$\mathrm{Pro}_{i,(\ldots,p_z,\ldots)} \xrightarrow{\mathrm{k_{d,z}}} \mathrm{Pro}_{i,(\ldots,0,\ldots)}, \tag{8}$$

$$\mathrm{Pro}_{i,(\ldots,p_z,\ldots)} + p_w \xrightarrow{\mathrm{k_{ijz,w}}} \mathrm{Pro}_{i,(\ldots,0,\ldots)} + p_z + p_w \tag{9}$$

$$RBS_i \xrightarrow{k_{dr,i}} \varnothing, \ p_i \xrightarrow{k_{dp,i}} \varnothing, \tag{10}$$

$$p_i + p_j \underset{k_{ij,un\,\mathrm{dim}}}{\overset{k_{ij,\mathrm{dim}}}{\rightleftharpoons}} p_{i,j} \tag{11}$$

For simplicity, proteins and RBS are assumed to degrade at a constant rate, modeled as uni-molecular reactions (reactions 10).

Ensembles of GRNs (Kauffman, 2004) can be generated by choosing random integers for all indexes in the reactions modeling interactions between genes (i, j, z and w).The choice of which dimers can form can also be random. Each different set of choices corresponds to a unique GRN topology. Since the effect of transcription factors in genes' expression level can be randomly chosen, the stochastic version of any Boolean or more complex transfer function can be implemented.

APPLICATIONS

Modeling Single Gene Expression

Recently, the real-time production of single protein molecules under the control of a repressed lac promoter in individual E. coli cells was monitored by epifluorescence microscopy (Yu et al, 2006). A model of this experiment is presented (Zhu et al, 2007), and can be used to validate our gene expression model.

In a constructed E. coli strain (SX4), a single copy of the chimeric gene tsr-venus was incorporated into the chromosome, replacing the native lacZ gene, while leaving intact the endogenous tsr gene. The introduced promoter is kept highly repressed, and thus is unable to express most of the time. Since the endogenous tsr gene expresses in high quantities, the addition of the extra gene (repressed most of time) doesn't affect the cell's normal behavior.

When the infrequent spontaneous dissociation of the repressor from the operator occurs, transcription begins. Usually this event generates a single mRNA, due to its short duration. When the mRNA is produced, a few ribosomes bind to it and proteins are produced. These can be detected after completion of their assembly process, which includes protein folding, incorporation onto the inner cell membrane, and maturation of the Venus fluorophore (Yu et al, 2006). Observing the radiation emission events, it was found that the proteins are produced in bursts, with the distribution of the bursts per cell cycle fitting a Poisson distribution, and that the number of proteins produced per burst follows a geometric distribution (Yu et al, 2006).

This example is used to validate our model of gene expression at the single-molecule level. The set of chemical reactions are the following:

$$\mathrm{Pro(t)} + \mathrm{RNAp(t)} \xrightarrow{\mathrm{k_{transc}}} \mathrm{Pro(t+\tau_1)} + \mathrm{RBS(t+\tau_1)} + \mathrm{RNAp(t+\tau_2)} + \mathrm{R(t+\tau_2)} \quad (12)$$

$$\mathrm{Ribosome(t)} + \mathrm{RBS(t)} \xrightarrow{k_{transl}} \mathrm{RBS(t+\tau_3)} + \mathrm{Ribosome(t+\tau_4)} + \mathrm{P(t+\tau_5)} \quad (13)$$

$$RBS \xrightarrow{RBSdecay} \varnothing \quad (14)$$

$$\mathrm{Pro(t)} + \mathrm{Rep(t)} \xrightarrow{\mathrm{k_{rep}}} \mathrm{ProRep\ (t)} \quad (15)$$

$$\mathrm{ProRep\ (t)} \xrightarrow{\mathrm{k_{unrep}}} \mathrm{Pro(t)} + \mathrm{Rep(t)} \quad (16)$$

Reactions 12 and 13 model, respectively, prokaryotic transcription and translation. R represents a complete RNA molecule and doesn't intervene in other reactions (it is included to allow an exact counting of the number of transcription events since no decay reaction is defined for R, thus, its quantity equals the total number of transcription events that occurred).

The RBS (ribosome binding site of the RNA) is the part of the RNA to which the ribosomes bind to and initiate the RNA translation. In prokaryotes, which is the case here, this can occur as soon as the RBS is produced (τ_1 seconds after the transcription event occurs). The RBS is subject to decay via reaction 14, avoiding the possibility of creating an infinite number of proteins, out of a single RNA.

Reaction 15 models the promoter repression by a repressor (Rep). Reaction 16 models the unbinding of the repressor from the promoter. Only when the promoter is free can transcription occur and, since

this reaction rate constant is very small, this occurs at very sparse intervals. The expected fraction of time that the promoter is going to be available for reactions is given by (17):

$$\left[1+\left(\frac{1}{k_{unrep}}+\tau\right).k_{rep}.\mathrm{Rep}\right]^{-1} \tag{17}$$

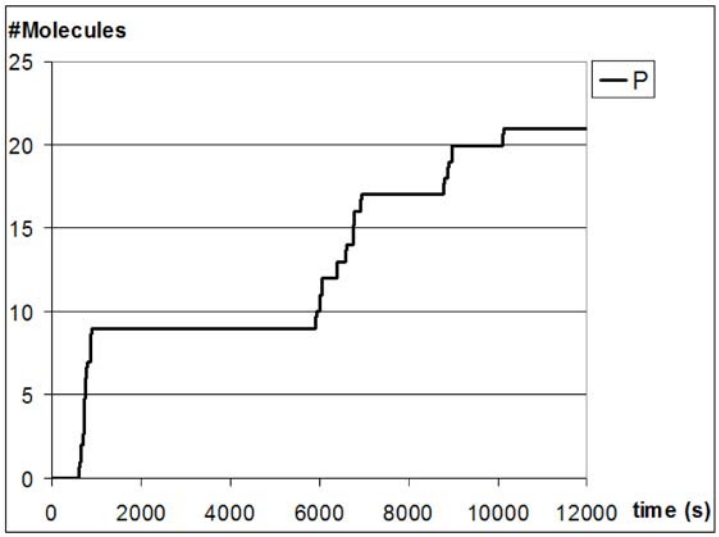
The rate constants are set for reactions 12-16 are: $k_{transc} = 0{,}01 s^{-1}$, $k_{transl} = 0{,}00042\ s^{-1}$, RBSdecay = 0.01 s^{-1}, $k_{rep} = 1\ s^{-1}$, and $k_{unrep} = 0{,}1\ s^{-1}$ Initially, initially it is set that RNAP = 40, Pro = 1, R = 0, Ribosome = 100, RBS = 0, P = 0, ProRep = 0, and Rep = 100. Time delays are set at: $\tau_1 = 40$ s, $\tau_2 = 90$ s, $\tau_3 = 2$ s, $\tau_4 = 58$ s, and $\tau_5 = 420 \pm 140$ s, randomly generated from a normal distribution of mean value 420 and standard deviation of 140 (with cutoff at 0).

In Fig. 1 is plotted the number of produced proteins. The proteins are produced in bursts, as reported (Yu et al, 2006). Each time the repressor unbinds the promoter, an RNAp can bind to the promoter, producing one RNA, which is translated into several proteins before decaying. The bursts in this simulation occurred at ~400, 6000 and 9000 seconds.

Running several simulations one observes that the moments the bursts occur and the number of resulting proteins from each event varies significantly, due to the stochastic nature of the dynamics. In Fig. 2A the number of transcription initiations distribution is plotted, over 1000 simulations. The resulting distribution of bursts size per cell cycle fits well a Poisson distribution.

In Fig. 2B, from the same 1000 simulations, it's shown that the number of translation reactions fits an exponential distribution, as reported in (Yu et al, 2006). Notice that an ODE model would not be able to reproduce the production by bursts, since it is not appropriate to model systems with very few molecules and where single events, sparse in time, are the relevant ones. Time delays also play an important role, limiting the number of RNAp molecules that can bind to the gene when unrepressed.

Figure 1. Time series of proteins production during 4 cell cycles of a single simulation



Bi-Stability of a Toggle Switch due to Delays in Transcription

The genetic toggle switch (TS) is among the most widely studied GRNs, due to its simplicity and important role as decision circuits in cell differentiation and as cellular memory units (Gardner et al, 2000). A TS is a 2-gene GRN in which each gene's proteins can bind to the other gene's promoter region, inhibiting its transcription.

The model of TS presented here (reactions 18-23) does not require protein dimerization or self-activation (Ribeiro, 2007). This model can reproduce the experimentally observed behavior of engineered TS's in real cells. Introducing realistic delays in the promoter release, at each transcription reaction, is sufficient to induce toggling. The model is used to show how relevant time delays are in the dynamics of even the simplest GRNs.

$$\mathrm{Pro}_1 + RNAp \xrightarrow{k_t} \mathrm{Pro}_1(\tau_1) + RNAp(\tau_2) + p_1(\tau_3) \tag{18}$$

$$\mathrm{Pro}_2 + RNAp \xrightarrow{k_t} \mathrm{Pro}_2(\tau_1) + RNAp(\tau_2) + p_2(\tau_3) \tag{19}$$

$$\mathrm{Pro}_1 + p_2 \underset{\mathrm{k_u}}{\overset{\mathrm{k_c}}{\rightleftharpoons}} \mathrm{Pro}_1 p_2 \tag{20}$$

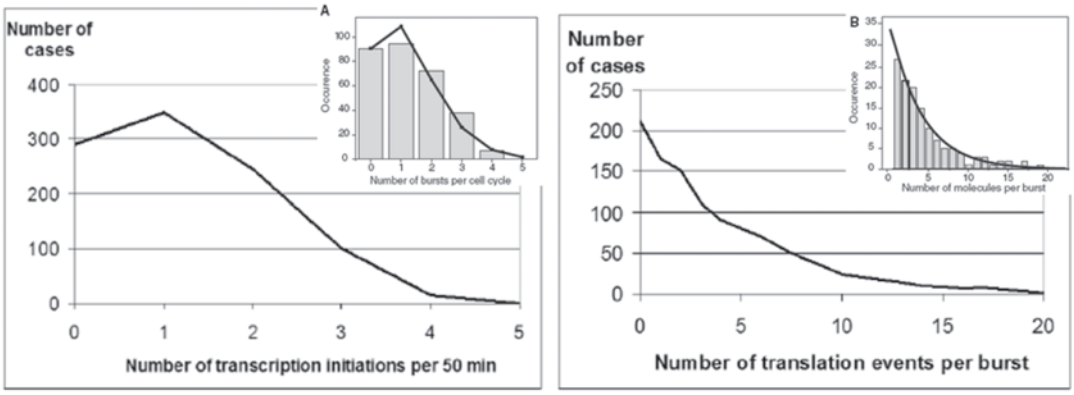
$$\mathrm{Pro}_2 + p_1 \underset{\mathrm{k_u}}{\overset{\mathrm{k_c}}{\rightleftharpoons}} \mathrm{Pro}_2 p_1 \tag{21}$$

$$\mathrm{Pro}_1 p_2 \xrightarrow{\mathrm{k_{dp}}} \mathrm{Pro}_1 \,,\ \mathrm{Pro}_2 p_1 \xrightarrow{\mathrm{k_{dp}}} \mathrm{Pro}_2 \tag{22}$$

$$p_1 \xrightarrow{k_d} \varnothing \,,\ p_2 \xrightarrow{k_d} \varnothing \tag{23}$$

Reactions 18 and 19 represent the transcription-translation of the genes in a single step, accounting for the time it takes on average for these two complex chemical processes to be finished once initiated. Reactions 20 and 21 control the coupling strength between genes of the two TS's, by setting the propensity for repressors to bind and unbind to the promoters. Reactions 22 and 23 are responsible for

Figure 2. (A) Number of transcription events during 4 cell cycles, in 1000 simulations. (B) Number of translation events for each RNA transcribed in 1000 simulations. The small figures show the experimental data for direct comparison.



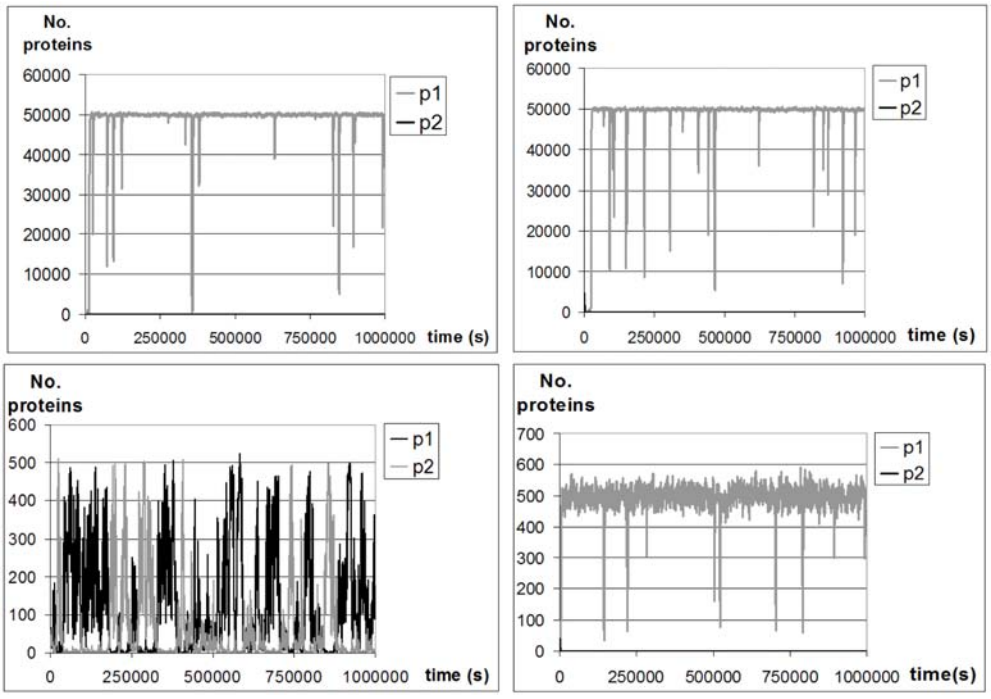
proteins decay. Reactions 22 allow the protein to decay when bound to the promoter at the same rate as when not bound. If absent, binding to the promoter would act as a protection against protein decay and affect the dynamics dramatically.

Given this set of reactions, we examine four cases: (A) no time delays, all τ's are set to null (Fig 3A); (B) a time delay only on the protein production, namely, $\tau_3 = 100$ s, while $\tau_1 = \tau_2 = 0$s (Fig 3B); (C) multiple delays, $\tau_1 = 2$s, $\tau_2 = 20$s and $\tau_3 = 100$s (Fig 3C); (D) same settings as (B) but with a transcription rate constant, ($k_t = 0.005\ s^{-1}$), 100 times smaller (Fig 3D).

In case A (no time delays), the TS does not toggle (Fig. 3A) since there is no cooperative binding (i.e., protein dimers as inputs to the promoters) and because there are no self-activation reactions (unlike in (Lipshtat, 2006)). After a transient, the TS settles into one of the two stable states (one gene on and the other off), each equally probable. The choice is driven by stochastic fluctuations. Once the choice is made, the system does not toggle anymore. The average transient is ~14000s with a standard deviation of ~7000s. Decay and production equilibrate at ~50000 proteins.

In case B, time delays for proteins' production are introduced, causing protein levels to fluctuate more but they still reach a single steady state, rather than toggling. The average transient time to attain the stable state increases to ~19000 s with a standard deviation of ~12500 s. After the transient, one of the genes becomes on at the level of 50.000 proteins, and the other off. The delays introduced only affect the initial transient and after that, the steady state solution is the same as if no delays existed.

Figure 3. Time series of TS: (A) without delays or cooperative binding. (B) 100 s delays on the proteins release and no cooperative binding. (C) multiple delayed transcription/translation and no cooperative binding. Delays: $\tau_1 = 2$ s (promoter), $\tau_2 = 20$ s (RNAp), $\tau_3 = 100$ s (proteins). (D) delays on the p's release only, and $k_t = 0{:}005\ s^{-1}$.



In case C, all delays are non-null. As seen in Fig. 3C, the system dynamics changes drastically in comparison with the previous two cases. First, the maximum level that proteins reach is 500 (in comparison with 50 000), due to the delay on the promoter that limits the number of RNAp molecules that can be transcribing the gene at the same time. Since $\tau 1 = 2$ s, there can be at most 1 transcription every 2 seconds.

The delay on the RNAp release also diminishes the transcription reaction propensity (approximately by 20%) since a fraction of the RNAp molecules is not available while occupied transcribing a gene. The system now toggles (from p1 being in larger quantity to p2 and vice-versa), after an average transient for the first toggling to occur of 4900 s with a standard deviation of 4050 s. The average number of toggles observed during the entire simulation is 18.5 with a standard deviation of 3.75. Thus, the average toggling period is 50 000 s.

The toggling observed in Fig. 3C either is due to the delay on the promoter or a consequence of having a far smaller number of proteins of each gene and thus stochastic fluctuations causes toggling (also indirectly caused by the delay on the promoter).

In case D, the toggling is caused by the delay on the promoter and not by having a small maximum number of proteins of the gene on. In this model, transcription/translation delays occur only on protein production, as in B, but with transcription rates 100 times smaller, so that the maximum level for the protein is the same as in C. For that we set $k_t = 0{:}01\ s^{-1}$. Such a decrease, as seen in Fig. 3D, sets the maximum number of proteins at ~500 as in C, but no toggling was ever observed. Thus, given no cooperative binding or self-activation, toggling is possible, and is caused by the delay on the promoter release.

Model of the P53-Mdm2 Network

The tumor suppressor protein P53 has a fundamental role in cellular response to a variety of environmental stresses that can affect DNA structure and replication. Depending on the causes of stress, P53 can activate several genes that regulate processes such as cell cycle arrest, DNA repair, and apoptosis (Volgstein et al, 2000). Mutations in the gene that transcribes p53 RNA have been found in about 50% of human tumors (Bennet, 1999).

We simulate a stochastic version of the P53-Mdm2 feedback loop that accurately matches recent experimental observations, namely that oscillations end abruptly at the single cells level (Ma et al, 2005), while when observing multiple cells, the oscillations resemble a damped oscillator. In addition, a time series of cells of a lineage at the single cell level (Geva-Zatorsky et al, 2006) can be accounted for by this model.

Under normal conditions, P53 concentrations are kept low by Mdm2 protein (Haupt et al, 2005). These two proteins form a negative feedback loop responsible for the oscillatory dynamics in their concentrations in cells exposed to radiation that induces DNA damage (Ma et al, 2005). When under stress, P53 concentration can rapidly increase by several folds.

Observations show that the number of DNA double strand breaks (DSBs) in the cells follows a Poisson distribution whose average is proportional to the radiation dose (Ma et al, 2005). DSBs are thus treated as a chemical species and inserted in the system at defined times, in a quantity randomly generated from a Poisson distribution.

The single cell model of the P53-Mdm2 consists of the following set of reactions (24-37) (refer to (Ribeiro et al, 2007a) for a complete description of reactions system):

$$DSB \xrightarrow{DSB_{decay}} \varnothing \tag{24}$$

$$\mathrm{ProP53} \xrightarrow{Sp53/5} \mathrm{ProP53}(\tau_{\mathrm{Pro53}}) + \mathrm{pr53}(\tau_{\mathrm{pr53}}) \tag{25}$$

$$\mathrm{ProP53} + {}^{*}\mathrm{DSB} \xrightarrow{4.Sp53} \mathrm{ProP53}(\tau_{\mathrm{Pro53}}) + \mathrm{pr53}(\tau_{\mathrm{pr53}}) \tag{26}$$

$$\mathrm{pr53} \xrightarrow{Gp53} \varnothing \tag{27}$$

$$\mathrm{pr53} \xrightarrow{rp53} \mathrm{pr53}(\tau^{RBS}_{\mathrm{Pro53}}) + \mathrm{P53}(\tau_{\mathrm{P53}}) \tag{28}$$

$$\mathrm{P53} \xrightarrow{up53} \varnothing \tag{29}$$

$$\mathrm{P53} + {}^{*}\mathrm{MDM2} \xrightarrow{up53} \varnothing \tag{30}$$

$$\mathrm{ProMdm2} \xrightarrow{Smdm2/10} \mathrm{ProMdm2}(\tau^{1}_{\mathrm{ProMdm2}}) + \mathrm{mdm2}(\tau_{\mathrm{mdm2}}) \tag{31}$$

$$\mathrm{ProMdm2} + {}^{*}\mathrm{P53} \xrightarrow{Smdm2} \mathrm{ProMdm2}(\tau^{2}_{\mathrm{ProMdm2}}) + \mathrm{mdm2}(\tau_{\mathrm{mdm2}}) \tag{32}$$

$$\mathrm{ProMdm2} + {}^{*}\mathrm{DSB} \xrightarrow{2.Smdm2} \mathrm{ProMdm2}(\tau^{3}_{\mathrm{ProMdm2}}) + \mathrm{mdm2}(\tau_{\mathrm{mdm2}}) \tag{33}$$

$$\mathrm{mdm2} \xrightarrow{Gmdm2} \varnothing \tag{34}$$

$$\mathrm{mdm2} \xrightarrow{rmdm2} \mathrm{mdm2}(\tau^{RBS}_{\mathrm{mdm2}}) + \mathrm{MDM2}(\tau_{\mathrm{MDM2}}) \tag{35}$$

$$\mathrm{MDM2} \xrightarrow{umdm2} \varnothing \tag{36}$$

$${}^{*}\mathrm{P53} + \mathrm{MDM2} \xrightarrow{vp532} \varnothing \tag{37}$$

The rate constants are set at: DSBdecay = 0.003 s^{-1}, Sp53 = 0.02 s^{-1}, Gp53 = 0.02 s^{-1}, Smdm2 = 0.045 s^{-1}, Gmdm2 = 0.02 s^{-1}, rp53 = 0.6 s^{-1}, up53 = 0.02 s^{-1}, vp53 = 9.2 s^{-1}, rmdm2 = 0.04 s^{-1}, and umdm2 = 0.14 s^{-1}. We set the time delays as: τ_{mdm2} = 100s, τ_{MDM2} = 10s, τ_{pr53} = 100s, τ_{P53} = 10s, τ_{ProP53} = 1s, $\tau^{1}_{ProMdm2}$ = 1s, $\tau^{2}_{ProMdm2}$ = 0.01s, $\tau^{3}_{ProMdm2}$ = 0.05s, τ^{RBS}_{mdm2} = 1s, and τ^{RBS}_{p53} = 0.1s. We also set the following initial quantities: P53 = 0, mdm2 = 0 (mdm2 RNA), MDM2 = 0, pr53 = 0 (p53 RNA), ProP53 = 1 (promoter region of the gene from which the p53 RNA is transcribed), and ProMdm2 = 1 (promoter region of the gene from which the mdm2 RNA is transcribed).

Reaction 33 models the activation of Mdm2 transcription due to the presence of DSB in the system, since it is known that when DSB exist, signaling molecules detect them and will then begin a cascade of events that will eventually lead to a higher expression of P53 and Mdm2 (Ma et al, 2005).

It was observed experimentally that the P53 and Mdm2 oscillations have an approximately constant frequency before stopping. The number of oscillations varies from cell to cell and, although a damped oscillation of P53 and Mdm2 is observed in the cell population average, in single cell measurements these oscillations are only slightly damped and appear to cease abruptly (Ma et al, 2005).

In Fig. 4A is shown the results of a single simulation. The results agree with the experiments (Ma et al, 2005) in the number of oscillations as a response to a single addition of DSB, in the P53 and Mdm2 relative peak intensity, and in the time interval between the peaks of the two substances. When DSB are introduced, P53 and Mdm2 can oscillate between 1 and 4 times. Also, the oscillations are damped and their ending is abrupt. Once the oscillations stop, only adding more DSBs can restart the oscillations.

The system responds diversely to each addition of DSBs, in amplitude and number of oscillations. Only the oscillations' period is almost invariable, in agreement with observations. Adding more DSBs originates a stronger response on average.

Next, 10 independent cells were simulated, all with the same initial conditions, except for the initial number of DSB, randomly drawn from a Poisson distribution. The average quantity of P53 in all 10 cells is shown in Fig. 4B. The average result is a damped oscillation although they differ significantly between individual cells.

Next, we modeled a cell line with 3 generations, created from an initial mother cell. Mother and daughter cells have the same set of possible chemical reactions (24-37). Only the mother cell is subject to an initial addition of DSB at t = 0 s.

The time series of the P53 protein of cells of the lineage are shown in Fig. 5, where one observes that the oscillations in mother cells continue in their daughter cells as the reported (Ma et al, 2006). Additionally, and also matching the measurements, as the cells are more distanced in the lineage, their dynamics differs more, both in phase and in amplitude. In some lines the oscillations have ceased while persisting in other lines.

Also observable is that as the two daughter cells of the same mother cell evolve in time, although their oscillations are perfectly correlated in the beginning, they lose correlation in both frequency and amplitude of oscillations (observe the time series of the second generation) (Ma et al, 2006).

EVOLUTION OF GENE REGULATORY NETWORKS

GRNs are generally complex, often consisting of highly interrelated connections that respond well to a wide range of environmental signals and conditions. The previous sections showed that models based on the SSA can successfully reproduce the dynamics of experimental systems. It is also important to ad-

Figure 4. (A) Time series of P53, Mdm2 and DSB's in a single cell. Sampling period is 10 s. DSB's are introduced at t=0 s. (B) Time series of P53 averaged over 10 independent cells.

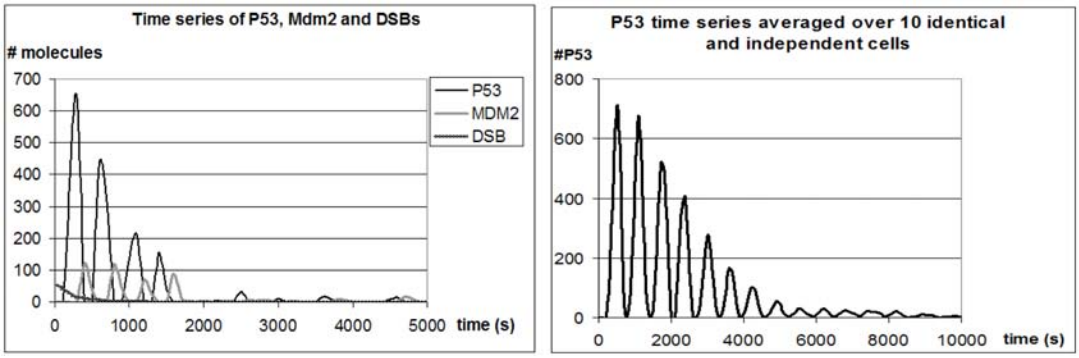
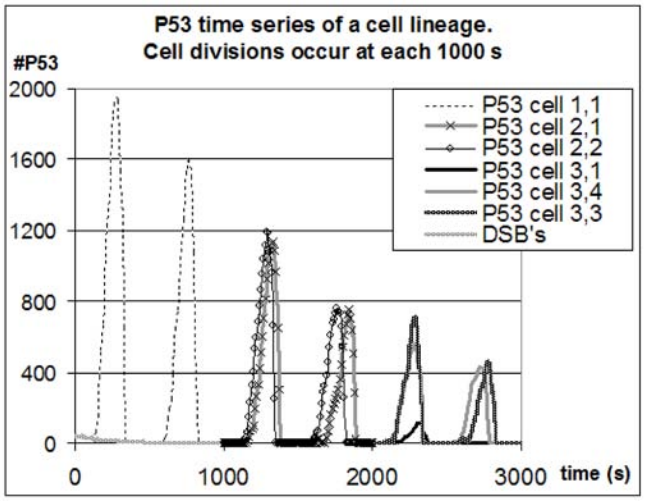


Figure 5. Time series of P53 in each cell of the cell line (except cell (3,2)) since its time series is very similar to cell (3,1)). At each 1000 s, cells divide and two daughter cells are created from each existing cell. Note that cells (3,3) and (3,4) have similar dynamics (almost synchronized in phase and amplitude) since they are daughters from the same mother cell (cell (2,2)), but are almost uncorrelated to cell (3,1) generated from a different mother cell (cell (2,1)). Sampling period is 10 s.



dress the evolvability of the control structure observed in biological systems. That is, a complete model of gene regulation should include an account of how observed regulatory structure may have evolved. A first step would be to examine how easy or difficult it would be to evolve various classes of genetic regulation. Toward this goal, we provide a brief discussion of how a genetic toggle switch might evolve assuming the SSA model described previously. While this work is admittedly speculative, it illustrates some possible important directions for further research in GRN simulation and modeling.

To explore the evolution of GRNs, we consider an abstract model of the biochemical mechanisms underlying regulatory relationships between genes, similar to the model presented in (Grefenstette et al, 2006). In the current model, a gene *i* is represented by two integers: Pro_i and p_i where Pro_i represents the gene's promoter site and p_i represents the gene's protein product. We use these numbers to model the affinity between regulatory proteins and promoter sites. In particular, we say that gene *j* binds to the promoter region of gene *i* promoter if $| p_j - Pro_i | < b_{thresh}$ (where b_{thresh} is the binding threshold) where the latter value is a parameter of the model. Assuming binding occurs, gene *j* activates gene *i* if $(p_j - Pro_i) \geq 0$, and gene *j* represses gene *i* otherwise.

We investigated the evolvability of toggle switches from a set of random networks, using a genetic algorithm (Grefenstette, 1986). We generated an initial population of 100 networks, each with 20 genes whose *Pro* and *p* values were generated at random in the range [0,5000]. In this study, $b_{thresh} = 100$. Given these parameters, the probability that any two genes would form a mutually binding pair is approximately 0.0016, but the probability of forming a successful toggle switch is far smaller. Each network was translated into a set of reaction rules as described above, and simulated three times for 500.000 s. We measure the "fitness" of a network by its ability to toggle between any pair of proteins. In particular, fitness was determined by:

a) the fraction of time when exactly one of the toggling proteins has quantity ≥ 100
b) the number of times the TS switches state within the simulation time period

Successive generations of GRNs were derived using a genetic algorithm: each GRN was replicated a number of times proportional to its fitness. After replication, each GRN was mutated by adding a small drift to the values of Pro_i and p_i, resulting in a perturbation of the probability of each protein binding to each promoter region. The question of interest is how often the TS evolve.

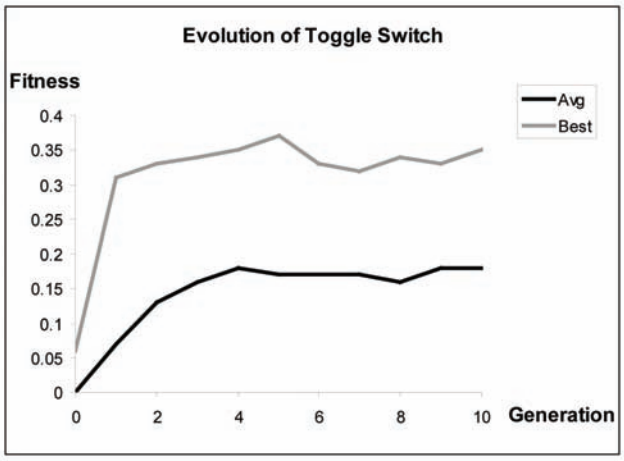
Fig. 6 shows a typical run of the evolutionary system, showing the average fitness of the entire population of networks as well as the fitness of the best TS in each generation. From these artificial evolutionary studies, it may be inferred that if the environment provides some selective advantage for toggling between two states of the GRN, it may be expected that TS will readily evolve.

Further studies are needed to explore the range of parameters under which TSs will evolve. It is also important to explore the evolution of TSs that respond to appropriate environmental signals (e.g., the presence or absence of nutrients). More generally, the evolution of complex GRNs remains an area for much future work.

CONCLUSION

The set of examples of GRNs in this chapter show the flexibility of this modeling strategy and the richness of the resulting dynamics. The simulations are computationally feasible, and the results match experimental observations at the highest level of detail. In addition, other aspects of GRNs dynamics, such as protein networks or microRNA, can be easily be incorporated in the model.

Figure 6. First 10 generations of a genetic algorithm. A population of 100 GRNs is evaluated for toggling behavior. Each GRN consists of 20 genes. Fitness is measured by the ability to toggle between states in which one of two proteins has high concentration. The lower curve shows the average fitness of the population; the upper curve shows the fitness of the best TS in each generation.



RESEARCH DIRECTIONS

Among many applications possible using this framework, one can test and refine inference algorithms of structure and logic of GRNs. Also, specific GRNs and chemical pathways can be modeled and effects of external perturbations measured. Many of these applications have direct medical implications and could prove valuable in the near future.

With respect to GRNs inference, some studies have been conducted to assert the ability of existing algorithms (originally designed for RBNs (Ribeiro et al, 2006b)) to infer models built with the modeling strategy here describe (Charlebois et al, 2008). The results yield some concern as to the ability of any existing inference algorithm, due to the noise of GRN dynamics and current level of noise of gene array technology or similar high-throughput data.

Since data on single cells is now coming available, examination of different cells of the same cell type across the entire genome for stochastic differences in expression may ultimately allow actual fitting of chemical master equations. We stress that no known experimental techniques allow high-throughput inference of genes' "transfer function". Even with respect to inputs, ChIP-chip is noisy, specific to cell types, and binding of a transcription factor does not assure its functionality in regulating transcription. The vastly more complex problem of understanding the "logic" of gene expression, given one to several inputs per gene, requires modulating the combinations of concentrations of these inputs and observing the output. For 25,000 genes in the human genome this is a huge task. Thus, inference may be an important tool in high-throughput analysis of gene transfer functions and a handmaiden to network topology as well.

In general, GRNs more complex than the TS are expected to have many "noisy attractors". Future work is needed to define more precisely in what sense localization of a system's dynamics in a small region of its state space constitutes a noisy attractor (Ribeiro et al, 2007c). More broadly, if cell types are noisy attractors, as hinted by metaplasia, then differentiation is either noise-induced or signal-induced transition between attractors, or bifurcations in which old attractors may disappear and new ones appear.

A fundamental problem is the directionality of ontogeny, i.e., starting with the zygote, differentiation is roughly a branching tree, or acyclic graph, with occasional cross connections. It is not at all clear why this should be so, for there is no known potential function for GRNs that would guide flow "downhill" from the zygote to terminal cell types. It remains a deep issue why ontogeny in differentiation is so largely unidirectional. Using delayed stochastic models of GRNs, and modeling cell types as noisy attractors, with noise induced differentiation among attractors, it appears to be a critical topic for future research to ask what classes of networks exhibit this "one way" property, and attempt to relate any success on this front to insight into the actual structure and logic of GRNs in multi-cellular organisms with one way ontogenies. One hopes that using inference methods and improved high-throughput gene expression static and time series data on single cells from distinct positions in branching cell differentiation lineages will aid in understanding these fundamental biological facts.

AKNOWLEDGMENT

We thank F.G. Biddle, Univ. of Calgary, for insightful discussions, Jason Lloyd-Price for invaluable contributions in the construction of the simulators, R. Zhu for insightful discussions, and S. Huang of the Univ. of Calgary, Canada, for useful discussions on the "one way" problem.

REFERENCES

Arkin, A., Ross, J., & McAdams, H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected E-coli cells. *Genetics, 149*, 1633–1648.

Becskei, A., & Serrano, L. (2000). Regulation of noise in the expression of a single gene. *Nature, 405*, 590–593. doi:10.1038/35014651

Bennett, W., Hussain, S., Vahakangas, K., Khan, M., Shields, P., & Harris, C. (1999). Molecular epidemiology of human cancer risk: Gene-environment interactions and p53 mutation spectrum in human lung cancer. *The Journal of Pathology, 187*(1), 8–18. doi:10.1002/(SICI)1096-9896(199901)187:1<8::AID-PATH232>3.0.CO;2-Y

Bratsun, D., Volfson, D., Tsimring, L. S., & Hasty, J. (2005). Delay-induced stochastic oscillations in gene regulation. *Proceedings of the National Academy of Sciences of the United States of America, 102*, 14593. doi:10.1073/pnas.0503858102

Charlebois, D., Ribeiro, A. S., Lehmussola, A., Lloyd-Price, J., Yli-Harja, O., & Kauffman, S. A. (2008). (accepted). Effects of microarray noise on inference efficiency of a stochastic model of gene networks. *WSEAS Transactions in Biology*.

Davenport, R., White, G., Landick, R., & Bustamante, C. (2000). Single-molecule study of transcriptional pausing and arrest by E. coli rna polymerase. *Science, 287*, 2497–2500. doi:10.1126/science.287.5462.2497

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science, 297*, 1183. doi:10.1126/science.1070919

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature, 403*, 339–342. doi:10.1038/35002131

Geva-Zatorsky, N., Rosenfeld, N., Itzkovitz, S., Milo, R., Sigal, A., & Dekel, E. (2006). Oscillations and variability in the p53 system . *Molecular Systems Biology, 2*. doi:. doi:10.1038/msb4100068

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry, 81*, 2340–2361. doi:10.1021/j100540a008

Grefenstette, J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-16*(1), 122–128. doi:10.1109/TSMC.1986.289288

Grefenstette, J., Kim, S., & Kauffman, S. (2006). An analysis of the class of gene regulatory functions implied by a biochemical model. *Bio Systems, 84*, 81–90. doi:10.1016/j.biosystems.2005.09.009

Haupt, Y., Maya, R., Kazaz, A., & Oren, M. (2005). Mdm2 promotes the rapid degradation of p53. *Nature, 387*, 296–299. doi:10.1038/387296a0

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology, 22*, 437–467. doi:10.1016/0022-5193(69)90015-0

Kauffman, S. A. (2004). A proposal for using the ensemble approach to understand genetic regulatory networks. *Journal of Theoretical Biology*, *230*(4), 581–590. doi:10.1016/j.jtbi.2003.12.017

Lipshtat, A., Loinger, A., Balaban, N. Q., & Biham, O. (2006). Genetic toggle switch without cooperative binding. *Physical Review Letters*, *96*, 188101. doi:10.1103/PhysRevLett.96.188101

Ma, L., Wagner, J., Rice, J., Hu, W., Levine, J., & Stolovitzky, G. (2005). A plausible model for the digital response of p53 to DNA damage. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(4), 14266–14271. doi:10.1073/pnas.0501352102

McAdams, H., & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *94*, 814–819. doi:10.1073/pnas.94.3.814

Mestl, T., Plahte, E., & Omholt, S. W. (1995). A mathematical framework for describing and analyzing gene regulatory networks. *Journal of Theoretical Biology*, *176*, 291–300. doi:10.1006/jtbi.1995.0199

Ota, K., Yamada, T., Yamanishi, Y., Goto, S., & Kanehisa, M. (2003). Comprehensive analysis of delay in transcriptional regulation using expression profiles. *Genome Inform.*, *14*, 302–303.

Raser, J. M., & O'Shea, E. K. (2004). Control of stochasticity in eukaryotic gene expression. *Science*, *304*, 1811–1814. doi:10.1126/science.1098641

Ribeiro, A. S. (2007). Effects of coupling strength and space on the dynamics of coupled toggle switches in stochastic gene networks with multiple-delayed reactions. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *75*(1).

Ribeiro, A. S., Charlebois, D., Lloyd-Price, J., & Kauffman, S.A. (2006, May 10-12). IADGRN: Inferring gene regulatory networks from time series of genes activity. Increasing the scope of efficiency to more general interaction functions between genes and more complex time series. *6th Int. Conf. Canadian Proteomics Initiative, CPI*, Edmonton, Alberta, Canada.

Ribeiro, A. S., & Kauffman, S. A. (2007). Noisy attractors and ergodic sets in models of genetic regulatory networks. *Journal of Theoretical Biology*, *247*(4), 743–755. doi:10.1016/j.jtbi.2007.04.020

Ribeiro, A. S., & Lloyd-Price, J. (2007). SGNSim, a stochastic genetic networks simulator. *Bioinformatics (Oxford, England)*, *23*(6), 777–779. doi:10.1093/bioinformatics/btm004

Ribeiro, A. S., Zhu, R., & Kauffman, S. A. (2006). A general modeling strategy for gene regulatory networks with stochastic dynamics. *Journal of Computational Biology*, *13*(9), 1630–1639. doi:10.1089/cmb.2006.13.1630

Roussel, M., & Zhu, R. (2006). Validation of an algorithm for delay stochastic simulation of transcription and translation in prokaryotic gene expression. *Physical Biology*, *3*, 274–284. doi:10.1088/1478-3975/3/4/005

Roussel, M. R. (1996). The use of delay differential equations in chemical kinetics. *Journal of Physical Chemistry*, *100*, 8323–8330. doi:10.1021/jp9600672

Süel, G. M., Garcia-Ojalvo, J., Liberman, L. M., & Elowitz, M. B. (2006). An excitable gene regulatory circuit induces transient cellular differentiation. *Nature*, *440*, 545–550. doi:10.1038/nature04588

Süel, G. M., Kulkarni, R. P., Dworkin, J., Garcia-Ojalvo, J., & Elowitz, M. B. (2007). Tunability and noise dependence in differentiation dynamics. *Science*, *315*, 1717–1719. doi:10.1126/science.1137455

Toulouse, T., Ao, P., Shmulevich, I., & Kauffman, S. A. (2005). Noise in a small genetic circuit that undergoes bifurcation. *Complexity*, *11*(1), 45–51. doi:10.1002/cplx.20099

Vogelstein, B., Lane, D., & Levine, A. (2000). Surfing the p53 network. *Nature*, *408*, 307–310. doi:10.1038/35042675

Yu, J., Xiao, J., Ren, X., Lao, K., & Xie, S. (2006). Probing gene expression in live cells, one protein molecule at a time. *Science*, *311*, 1600–1603. doi:10.1126/science.1119623

Zhu, R., Ribeiro, A. S., Salahub, D., & Kauffman, S. A. (2007). Studying genetic regulatory networks at the molecular level: Delayed reaction stochastic models. *Journal of Theoretical Biology*, *246*(4), 725–745. doi:10.1016/j.jtbi.2007.01.021

KEY TERMS AND DEFINITIONS

Gene Regulatory Network: also called a GRN or genetic regulatory network) is a collection of DNA segments in a cell which interact with each other (indirectly through their RNA and protein expression products) and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA.

Dynamical Behavior: Also called dynamic temporal behavior, is the trajectory of states, in a state space, followed by a system during a certain time interval.

Stochastic: Means "random". In the present chapter, the term stochastic describes the system's behavior, as "non-deterministic", because of the effects of noise in the system's dynamical behavior, that cannot be pre-determined.

Differentiation: Also known as "cellular differentiation", the process by which a cell type becomes another cell type, usually more specialized.

Time Delay: Time interval between the occurrence of a chemical reaction, and the appearance of the products of that reaction in the system.

Transcription: The process of copying DNA to RNA by an enzyme called RNA polymerase (RNAp).

Translation: Translation is the first stage of protein biosynthesis (part of the overall process of gene expression). Translation is the production of proteins by decoding mRNA produced in transcription.

Evolution: Evolution is change in the inherited traits of a population of organisms from one generation to the next. These changes are caused by a combination of three main processes: variation, reproduction, and selection. Genes that are passed on to an organism's offspring produce the inherited traits that are the basis of evolution. These traits vary within populations, with organisms showing heritable differences in their traits. When organisms reproduce, their offspring may have new or altered traits. These new traits arise in two main ways: either from mutations in genes, or from the transfer of genes between populations and between species. In species that reproduce sexually, new combinations of genes are also produced by genetic recombination, which can increase variation between organisms. Evolution occurs when these heritable differences become more common or rare in a population.

Chapter 9
Nonlinear Stochastic Differential Equations Method for Reverse Engineering of Gene Regulatory Network

Adriana Climescu-Haulica
Université Joseph Fourier, France

Michelle Quirk
Los Alamos National Laboratory, USA

ABSTRACT

In this chapter, we present a method to infer the structure of the gene regulatory network that takes in account both the kinetic molecular interactions and the randomness of data. The dynamics of the gene expression level are fitted via a nonlinear stochastic differential equation (SDE) model. The drift term of the equation contains the transcription rate related to the architecture of the local regulatory network. The statistical analysis of data combines maximum likelihood principle with Akaike Information Criteria (AIC) through a forward selection Strategy to yield a set of specific regulators and their contribution. Tested with expression data concerning the cell cycle for S. Cerevisiae and embryogenesis for the D. melanogaster, this method provides a framework for the reverse engineering of various gene regulatory networks.

INTRODUCTION

A foreground question towards the understanding of the cell regulatory mechanism is that on how to infer the structure of the transcriptional regulatory network from experimental data.

To answer this question equates to decipher the learning machinery which enables the transcriptional program to adapt in time as the cell progresses through development or undergoes environmental changes. A current trend to trace dynamic features on the relationship between genes and their regulators is to analyze time-dependent microarray gene expression data obtained in pertinent conditions. The quantitative

DOI: 10.4018/978-1-60566-685-3.ch009

analysis of the variation of mRNA levels is expected to reverse-engineer the transcriptional regulatory network architecture, once this quantitative analysis is corroborated by qualitative tools to recognize specific promoter sequences, binding sites and transcription factors. Novel computational strategies arise from this perspective and a first major impact is expected in disease control.

The modeling methodology for regulatory networks has to demonstrate awareness of the quickly moving perspective in molecular biology. Recent research - only four years after the completion of the Human Genome Project - reveals that the human protein coding of genes resumes to a smaller set accounting for only 20500 (Clamp et al., 2007).

This conclusion emerges from the fully acquired - by now - observation that the majority of the transcriptional output of the genomes of higher organisms is noncoding RNAs (Claverie, 2005). It is assumed that noncoding RNAs are the key to the genetic control architecture as a very complex system for cis- and trans-acting RNA based regulatory network and gene-gene communication via RNA-DNA/chromatin, RNA-RNA and RNA-protein interactions. Hence, the understanding of the regulatory network invokes the study of epigenomic phenomena. The protein coding gene lies in a very plastic environment able to learn and adapt to different conditions by means of a large panel of mechanisms. The experiment traceability of these mechanisms is still a work in progress; however an integrative system biology approach is sought to combine different layers of information available from

1. the production of various types of experimental data (microarray, combined microarray (He et al., 2006), DNA microarray (Tavazoie et al., 1999), ChIP-chip(Ren et al., 2000) ;
2. the results obtained from processing the data with different computational approaches ;
3. the genetic, molecular and biochemical studies.

Accordingly, the computational methods for regulatory network inference have to be built in a robust evolutive manner, to allow the assimilation of novel discoveries.

The method proposed in this chapter renders a framework which may adapt to different types of gene expression data. An automated procedure is given; it takes as input a gene expression data set and an ensemble of candidate regulatory genes, considered from up to date discoveries. The output provides the structure of the gene regulatory network, expressed as a list of potential **activators** and **repressors** for each gene of the input data set.

The following section shows the principal characteristics of this method in light of the actual research in the area of gene regulatory network inference from expression data. The first part of the main thrust of the paper describes the construction of the nonlinear SDE used to model the dynamics of a target gene expression level together with the statistical analysis and the corresponding algorithm. The second part shows that, applied to the expression measurements of the mRNA levels of *Saccharomyces cerevisiae* (Spellman et al., 1998), this model improves the fitting results from previous studies. We provide also the analysis of time dependent gene expression measurements on Drosophila melanogaster embryogenesis (Tomancak et.al, 2002).

Our goal is to provide tools for large scale investigation of transcriptomic data – thus we describe an improved method able to extract information on the cell regulatory mechanism, and potentially to contribute to the reverse engineering of the transcriptional regulatory network.

BACKGROUND

Several quantitative methods have been proposed to describe the causal relationships between the mRNA expression levels of a target gene and its potential regulators. The correlation level between genes expression intensities is not, however, a criteria to identify correctly the regulators (Li et al., 2004). Based on promoter regions analysis, an insightful study (Lee et al, 2002) reveals the relationship between the target gene and regulators via a set of network motifs. These motifs suggest a regulatory mechanism in terms of autoregulation, multicomponent loops, feedforward loops, single-input, multi-input, and a regulator chain.

Inspired from electrical networks architecture, these models are meritorious – yet they do not imply the flexibility characterizing the transcriptional regulatory networks. The transcriptional regulatory network function depends on both qualitative and quantitative aspects; for example Guet at al. in 2002 show how differences in quantitative reaction rates have drastic effects on the function of the circuits with identical qualitative organizational properties as connectivity and logic.

Alternative methods develop strategies aimed to fit a mathematical model, using qualitative and quantitative elements, with a set of putative regulators to estimate the transcription pattern of a specific target gene. Several types of differential equations (Vu et al. 2007; Novikov et al., 2008) and stochastic differential equations (Chen et al., 2005; Climescu-Haulica and Quirk, 2007) are examples of mathematical models with good results toward identifying the regulators and predicting their function as **activators** or **repressors**. The variety of such models reinforces the idea that there is no unique pattern of regulation: each gene or category of genes may have its own quantitative/qualitative design. Thus, it is noteworthy to investigate new models able to infer unknown regulatory patterns.

Three attributes characterize the mathematical model we propose in this study for the processing of time dependent gene expression data to detect transcriptional regulators and to estimate their level of contribution:

1. It is built on a probabilistic framework to embed random variations of the microarray data; a Brownian Motion process models the noise term, taking into account the superposition of small random factors that arise dynamically in time.
2. For each target gene there is a choice for the prototype of the regulatory function between the beta sigmoid function - designed to keep track of the local temporal patterns of the target gene regulators - and the sigmoid function which is shaped around statistical parameters; this feature accommodates partially the variability of the regulatory pattern from one gene to another.
3. It considers a kinetic interaction model for the decay rate accounting for the mRNA degradation; this leads to a nonlinear representation of the stochastic differential equation which models the target gene mRNA.

In this setting the stochasticity is modeled at two levels: the measurement random error of data and the randomness of the biological phenomena.

Although the Bayesian methods that infer regulatory networks take in account the randomness of the biological system, the SDE method has the advantage that it can retrieve the network feedback loops.

By comparison with the module network method (Segal et al., 2003) the stochastic method proposed in this chapter shares the same logic in the choice of the input candidate set of regulators yet it delivers a list of regulators corresponding to a single gene rather than to an entire module. This reflects better the

observed phenomena since the regulatory relationships are mostly specific to a regulator and its target and cannot be spread to an entire module.

The nonlinear SDE method may evolve quickly in order to accommodate data for different organisms and various experiments. It offers a scheme of work to be developed in conjunction with new insights and discoveries.

SDE TRANSCRIPTIONAL REGULATION MODEL

The idea of using SDE as a model of the temporally dynamic gene-transcription process arises naturally as there are at least two indications of stochasticity in the measured mRNA expression level. At the first sight we note already that the variation with time of the mRNA expression levels seems „chaotic" from one measured gene to another. Secondly, it is acknowledged that genetically identical cells exposed to the same environmental conditions can show significant variation in molecular content (Kaern et al., 2005). This variability is linked to stochasticity in gene expression and necessitates an appropriate model.

The framework of the SDE theory is technically adapted to describe the stochastic dynamics of the target mRNA expression level because it accommodates stochastic processes on both, drift and **noise** terms. In our model the drift term of the SDE depends on the regulation rate of the target gene. This is the main part of the equation where the regulatory network relationships are represented. Precisely, the regulation rate is modeled as a linear combination of the regulatory functions of network elements to be identified. We present the results obtained using two prototypes of regulatory functions: the sigmoid function as given by Chen et al. in 2005 and a beta sigmoid function we proposed previously designed to keep track of the local temporal patterns of the target gene regulators. The **noise** term is modeled by a Brownian Motion process which accounts for the superposition of small random factors that arise dynamically. In our model the **noise** is seen as the part which is „non structurable" in the network architecture to be explored.

SDE Rationale

Let T denote a discrete set that corresponds to the time instants of the gene expression measurements. Consider two stochastic processes defined for a given target gene, $(N_t)_t \in \mathrm{T}$ and $(X_t)_t \in \mathrm{T}$ that model, respectively, the variation in time of the target gene amount of mRNA and the variation in time of the expression level of mRNA. Let be the set of potential regulators for the target gene. Denote by g_t the function that models the transcription rate of the target gene at time t

$$g_t: P(\mathrm{R}) \rightarrow \mathrm{R}_+$$

where $P(\mathrm{R})$ is the set of all possible subsets of R and R_+ is the set of real positive numbers. Denote the real, positive mRNA degradation rate by a function of time $\lambda(t)$. We assume that the mRNA degradation effect is modeled by the kinetic equation of a first order chemical reaction

$$\lambda(t) = \lambda N_t$$

The model we proposed assumes that from time t to $t+\Delta t$ the transcription and degradation process are given by

$$\frac{N_{t+\Delta t} - N_t}{N_t} = (g_t - \lambda N_t)\Delta t + \sigma \Delta W_t$$

where $(W_t)_t \in \mathrm{T}$ is a Brownian Motion stochastic process restricted to the discrete index set T. This process models the random error and σ is a positive scaling parameter. At the infinitesimal time intervals, when $\Delta t \to 0$ the above equation becomes a **stochastic differential equation**

$$\frac{dN_t}{N_t} = (g_t - \lambda N_t)dt + \sigma dW_t$$

The **stochastic differential equations** are very different from the ordinary differential equations. Firstly, because the Brownian Motion process which drives a SDE is not differentiable in the usual sense. It requires its own rules of calculus which forms the object of the theory of stochastic calculus (see for example the book of Karatzas et Shreve, 1991.) In a second place, the solution of a SDE is a stochastic process. This fact happens to be appropriate to many real applications where time varying data show random variations. In particular, a Brownian Motion stochastic process $(W_t)_t$ is a good model of the random **noise** as it can be seen from its characterizing properties:

1. W_0=0;
2. Almost all the paths of the stochastic process $(W_t)_t$ are continuous;
3. For each $0 \le s < t < u < v$, the increments W_t-W_s and W_v-W_u are independent; each increment W_t-W_s is distributed as $N(0,t-s)$.

Since N_t is proportional with the signal intensity S_t, and $X_t = \log(S_t - B)$ – where B is the background intensity – assume without loss of generality that

$X_t = \log(N_t)$.

Thus, the Itô chain rule of the stochastic calculus applies (Karatzas et Shreve, 1991) and the SDE obtained for X_t yields

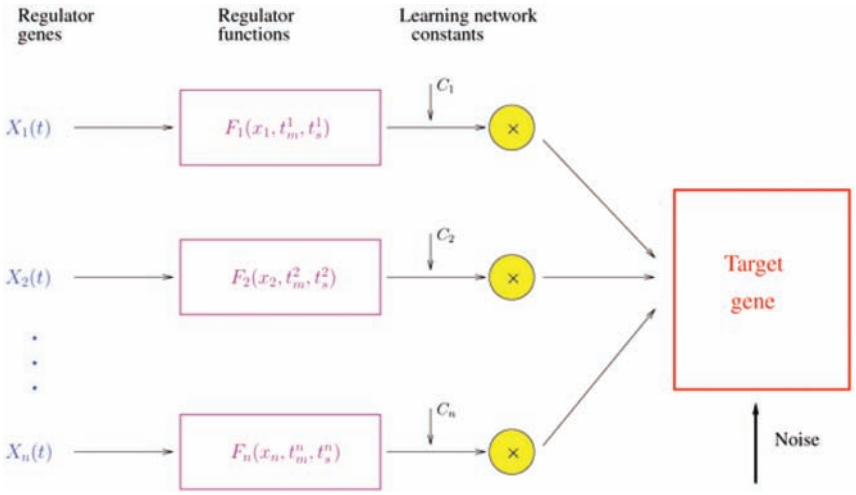
$$dX_t = (g_t - \lambda \exp X_t - \frac{\sigma^2}{2})dt + \sigma dW_t \qquad (1)$$

This is a nonlinear SDE for which the **drift** terms takes in account the temporal contribution of the kinetic interactions on the mRNA degradation.

Local Regulatory Network

In the above equation the informative part about the network structure is contained in the **drift** term, given by g_t, the function modeling the transcription rate. Consider an increasing sequence of temporal values

Figure 1. The picture of the regulatory network architecture in the neighborhood of the target gene



$T = \{t_0 < t_1 < ... < t_n\}$

Let m be the cardinality of the set R and let X_t^i be the mRNA expression level of the i-th regulator from the set R, measured at time t from T. Denote

$$\underline{X}^i = (X_{t_0}^i, X_{t_1}^i, ..., X_{t_n}^i)$$

In the neighborhood of the target gene the regulatory network is represented locally as a superposition of regulatory elements. The causal relationship between the target gene and its regulators, depicted in figure 1, is modeled as an indirect learning relationship. This local network relationship is expressed by the transcription rate function, built from the observable information, i.e. the regulators mRNA expression levels, as:

$$g_t = c_0 + \sum_{i=1}^{m} c_i F_i(\underline{X}^i, t) \tag{2}$$

where F_i denotes the regulatory functions of the potential regulators from R. The constants c_0, c_1,...,c_m are the learning parameters of the network. They modulate the network behavior and carry information in both their magnitude and sign about the local regulatory process: positive values correspond to regulators with activation, and negative values correspond to repression.

The learning in the local network is driven by the nonlinear SDE

$$dX_t = \left[\bar{c}_0 + \sum_{i=1}^{n} c_i F_i(X_t^i) - \lambda \exp X_t\right] dt + \sigma dW_t \tag{3}$$

where $\bar{c}_0 = c_0 - \sigma^2 / 2$.

Sigmoid and Beta Sigmoid Patterns of Regulation

The regulatory function is central to the model and fits the quantitative pattern with a specific regulator that acts on the mRNA expression of the target gene.

Our work investigates two prototypes of the regulatory function. We analyze on the nonlinear SDE framework the prototype of regulatory function introduced by Chen et al. in 2005, defined by

$$\eta(\underline{X}^i, t) = \frac{1}{1 + e^{-(X_t^i - \mu_i)/\sigma_i}} \tag{4}$$

where μ_i and σ_i are the mean and deviation of $\underline{X}$i as well as the prototype of the regulatory function we proposed previously, based on the beta sigmoid function expressed by

$$\beta(\underline{X}^i, t) = x_{\max}^i \left[1 + \frac{t_m^i - t}{t_m^i - t_s^i}\right]\left(\frac{t}{t_m^i}\right)^{\frac{t_m^i}{t_m^i - t_n^i}} \tag{5}$$

In this expression $x_{\max}^i$ represents the maximal value corresponding to mRNA expression levels of the potential regulator i

$$x_{\max}^i = \max\left\{X_t^i \mid t \in T\right\}$$

t_m^i is the first time when the maximum is attaint

$$t_m^i = \min\left\{t \in T \mid X_t^i = x_{\max}^i\right\}$$

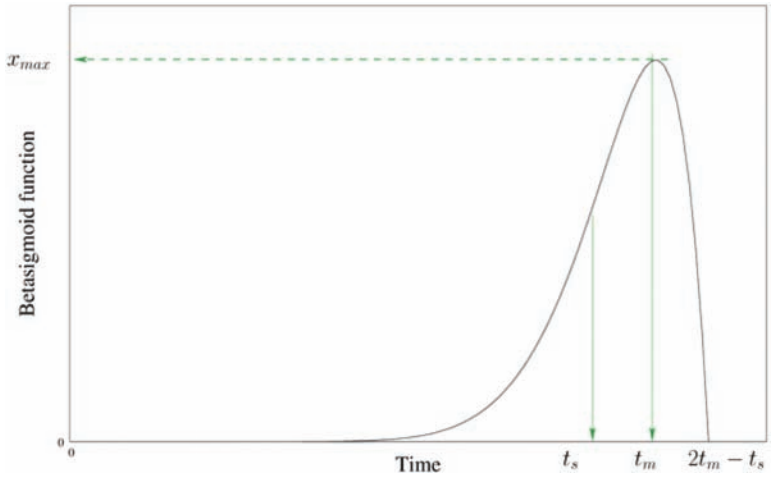
and t_s^i corresponds to the time for which the increase of the mRNA expression levels become maximale

The sigmoid and the betasigmoid functions are very different in shape and meaning. The sigmoid function uses only statistical parameters, taking in account the average behavior of the regulator gene. The beta sigmoid function keeps track of the temporal characteristics of the regulator gene. The parameter t_s^i corresponds to the point where the regulator expression level begins to increase. The maximal contribution of the regulator *i* is induced in the target gene at time t_m^i, when the mRNA expression level of the regulator attends its maximum, corresponding to the biological hypotheses.

Figure 2 shows an example of beta sigmoid function shape. The beta sigmoid function degenerates after time $2\,t_m^i$ - t_s^i and becomes non-informative. For this the effective expression of the regulatory function is given by

$$F_i(\underline{X}^i, t) = I_{\left\{\beta(\underline{X}^i, t) > 0\right\}}\left(\underline{X}^i, t\right)\beta\left(\underline{X}^i, t\right) + I_{\left\{\beta(\underline{X}^i, t) \le 0\right\}}\left(\underline{X}^i, t\right)\eta\left(\underline{X}^i, t\right) \tag{6}$$

Figure 2. Example of shape for the beta sigmoid function



where $\mathbf{I}_A$ is the indicator function of the set A ($\mathbf{I}_A(x) = 1$ if $x \in$ A and $\mathbf{I}_A(x) = 0$ if $x \notin$ A).

In was shown previously (Climescu-Haulica and Quirk, 2007) that this model of the regulatory function improves the fitting results obtained with the sigmoid function for almost 30% of the considered gene set. Therefore, the idea that there is no unique regulatory pattern of regulation directs us to consider both models.

STATISTICAL ANALYSIS

For a given target gene, the aim of the statistical analysis is to extract from the time course mRNA levels

1. the set of *m* regulators (model selection);
2. their corresponding parameters σ and the set $\{\lambda, c_0, c_1,...,c_m\}$ of parameters estimation;

where the best fit with respect to the nonlinear SDE is expressed in equation 3. The sigmoid and beta sigmoid as regulatory functions add supplementary parameters (μ_i , σ_i, t_s^i , t_m^i and $x_{\max}^i$ **)** to the model. These parameters are estimated from the corresponding time course mRNA levels according to their definitions and employed in the computation of the estimators of σ, λ and $\underline{\mathbf{c}}$:

The statistical procedure followed in this study is derived from the maximum likelihood principle (Casella and Berger, 2001) in combination with the Akaike Information Criterion (Akaike, 1974).

The regulators are considered as predictors in statistical sense, and the target gene is regarded as a response variable. The statistical approach is to fit the nonlinear model using a set of regulatory functions of regulators as the inputs to estimate the dynamic transcription rate of a target gene as the output. It also estimates the contribution and regulatory abilities of selected regulators. Equation (3) is considered in discrete form for each time interval $[t_j, t_{j+1}]$, $j = \{1, 2,...,n\}$ that corresponds to time measurements:

$$\Delta X_t = \left| c_0 + \sum_{i=1}^{n} c_i f_i\left(X_{it}\right) - \lambda \exp X_t \right| \Delta t + \sigma \Delta W_t$$

Then

$$\frac{\Delta X_t}{\sqrt{\Delta t}} = \left| c_0 + \sum_{i=1}^{n} c_i f_i(X_{it}) - \lambda \exp X_t \right| \sqrt{\Delta t} + \sigma Z_t$$

where the basic properties of Brownian Motion are considered in the expression of $Z_t = \Delta W_t / \sqrt{\Delta t}$:the increments ΔW_t are pairwise independent and each increment is normally distributed, with zero mean and standard deviation given by $N(0, \sqrt{\Delta t})$. Hence, $(Z_{t_j})_j$ form a family of i.i.d. random variables *N(0,1)* distributed.

Thereafter, for a specific target gene regulated by *n* regulators, *m* samples are collected at time $t = t_1, t_2, \ldots, t_m$, we have

$$\frac{\left(X_{t_{j+1}} - X_{t_j}\right)}{\sqrt{t_{j+1} - t_j}} = c_0 \sqrt{t_{j+1} - t_j} + \sum_{i=1}^{n} c_i \sqrt{t_{j+1} - t_j} f_i(X_{it_j}) - \lambda \exp X_{t_j} \sqrt{t_{j+1} - t_j} + \sigma Z_{t_j} \tag{7}$$

for $j = 1, 2, \ldots, m-1$. Let

$$Y_j = \frac{\left(X_{t_{j+1}} - X_{t_j}\right)}{\sqrt{t_{j+1} - t_j}} \tag{8}$$

$$U_j = \left[\sqrt{t_{j+1} - t_j}, \sqrt{t_{j+1} - t_j} f_1(X_{1t_j}), \ldots, \sqrt{t_{j+1} - t_j} f_n(X_{nt_j}), \exp X_{t_j} \right]$$

$$C = \left[c_0, c_1, \ldots, c_n, -\lambda \right]^T$$

then

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \cdots \\ Y_{m-1} \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ \cdots \\ U_{m-1} \end{bmatrix} C + \sigma \begin{bmatrix} Z_{t_1} \\ Z_{t_2} \\ \cdots \\ Y_{t_{m-1}} \end{bmatrix}$$

For simplicity, the above model is written in the matrix notation as $Y = UC + \sigma Z$ where ***Y***, ***Z*** are $(m-1)$ x 1 vectors and ***U*** is a $(m-1)$ x $(n+2)$ matrix. Moreover, ***U*** is the observed input, and ***Y*** is the observed output; ***C*** and σ are the parameters to be estimated.

Maximum Likelihood Principle As Statistical Estimation Method

We employ the **Maximum Likelihood** principle as method to estimate these parameters. Our sample is given by y=$(y_1,y_2,\ldots y_{m-1})$.The likelihood function *L* associated with this sample is defined as the probability that the random vector **Y** takes the value $(y_1,y_2,\ldots y_{m-1})$, i.e $L = P(Y_1 = y_1, Y_2 = y_2, ..., Y_{m-1} = y_{m-1})$.

Because **Y** depends on the parameters *C* and *σ,* the likelihood L is a function on the set of all possible values these parameters may take. The Maximum Likelihood principles states that the "most likely" values for the parameters *C* and *σ* are the values which maximize the likelihood function. Therefore the Maximum Likelihood estimators for *C* and *σ* are defined as

$$(\hat{C}, \hat{\sigma}) = \arg\max_{C,\sigma} \quad L(C, \sigma).$$

In many practical situations the log-likelihood function is preferred because it preserves the maximum and its computation is less complex. In our case, since $(Z_{t_j})_j$ are i.i.d. random variables with standard normal distribution, the log-likelihood function of ***Y*** comes from the formula

$$\log L = \log\left(\prod_{j=1}^{m-1} P(Y = y_j)\right) = \sum_{j=1}^{m-1} \log P(Y = y_j)$$

and is obtained from a standard computation for normal random variables as

$$\log L = -\frac{m-1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(y - UC\right)^T\left(y - UC\right)$$

Since the second-order partial derivative is negative,

$$(\partial^2 \,/\, \partial(\sigma^2)^2)\log L\,|_{C=\hat{C},\sigma^2=\hat{\sigma}^2} < 0$$

the MLE estimators for the parameters *C* and σ are obtained from the system formed by the two equations $(\partial \,/\, \partial C)\log L\,|_{C=\hat{C},\sigma^2=\hat{\sigma}^2} = 0$
And

$$(\partial \,/\, \partial\sigma^2)\log L\,|_{C=\hat{C},\sigma^2=\hat{\sigma}^2} = 0$$

which yields

$$\hat{C} = (U^T U)^{-1} U^T y \tag{9}$$

$$\hat{\sigma}^2 = \frac{\left(y - U\hat{C}\right)^T\left(y - U\hat{C}\right)}{(m-1)} \tag{10}$$

By the functional invariance property of ML estimators $\hat{\sigma}$ and $\hat{C}$, the estimated log-likelihood function is given by the formula

$$\log \hat{L} = \log L(\hat{\sigma}, \hat{C}),$$

and becomes, by replacing the expressions from the relation (9) and (10)

$$\log \hat{L} = -\frac{m-1}{2}\log(2\pi\hat{\sigma}^2) - \frac{m-1}{2} \tag{11}$$

Note that the ML statistical procedure estimates the degradation parameter λ contained in the last term of the vector C.

A Model Selection Procedure from Akaike Information Criterion

For the computation of the **ML** estimators we assumed the presence in our model of a number of *m* regulators which contribute to the *m-1* dimensional random vector **Y**. How to decide how many regulators to take in consideration for each target gene and which ones is not a trivial problem. This type of question is addressed by mathematical techniques of model selection. We consider a procedure of the model selection based on **Akaike Information Criterion** (**AIC**). This criterion states that between any two combinations of regulators, the best combination is such that the AIC of the regulators has the smallest value. The formula proposed in 1974 by Akaike for this criterion is expressed as

$$AIC = -2\log \hat{L} + 2(m+1) \tag{12}$$

where $\log \hat{L}$ is the estimator of log*L* computed previously and *m+1* is the number of independent parameters used for the computation of the **ML** estimators $\hat{\sigma}$ and $\hat{C}$ (the number of regulators and the time). The considerations which leaded to the AIC is related with the fact that the ML estimators are asymptotically efficient. For a not too large sample the reflex is to minimize the Kullback-Leibler distance between the probability density function of the sample and the parametric family of density functions considered for the computation of the **ML** estimators. Using statistical arguments and simplifications on the computation of this distance, Akaike proposed formula (12) to minimization criterion.

This tool provides a good fit of the dataset and is largely used as it offers quality results for a computation not costly in complexity.

Computational Algorithm

The objective of the computational work is to select a set of possible regulators to estimate the dynamic expression level of a target gene. Let H denote the set formed by a candidate pool of regulators of the target gene; denote by |H| the cardinality of H. Ideally, **ML** and **AIC** procedures shall be performed on each combination of regulators from H. Since the number of all possible combinations of regulators is $2^{|\mathfrak{H}|}$, an enumeration algorithm for those sets will explode quickly. The heuristic procedure used is the **forward selection** strategy (Weisberg, 1985, chapter 8). At first the regulator with the biggest log-likelihood with respect to the target gene is selected. A new regulator is added if it will increase the **AIC** more than any other single regulator outside the current combination. The actual implementation stops for a combination of maximum 10 regulators. A preprocessing step is needed in order to fulfill the eventual missing

data. Our estimation of missing value uses the interpolation of the adjacent time-point observations. If the missing value is at the first or the last time point, an extrapolated estimate is applied.

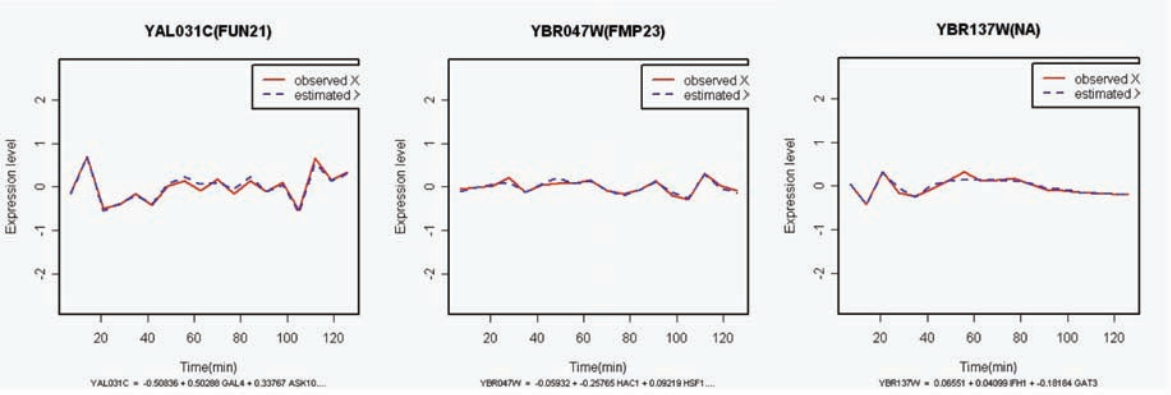
A as a validation step, we applied a least square approach to post-process the results. The winning combination of regulators renders large log likelihood, small AIC and small square errors between the expression value of the target gene and its estimated value. The main steps of the complete algorithm are the following

- *Step 1*. If the dataset is incomplete apply a missing data method (interpolation/extrapolation).
- *Step 2*. Estimate the statistical and temporal parameters corresponding to the sigmoid and beta sigmoid functions for all genes in the candidate pool and calculate the regulatory functions according to equations (4) and (6).

For a given gene in the data set perform the following:

- *Step 3*. Calculate the response ***Y*** by equation (8)
- *Step 4*. For each gene in the candidate pool of regulators estimate the vector of parameters C and sigma corresponding to the model given by equation (7), using relations (10) and (11).
- *Step 5*. Initiate the forward selection procedure: choose a single predictor (regulator) that has the biggest log-likelihood with the response (target gene).
- *Step 6*. Add an additional predictor from the candidate pool of regulators if it meets the following criteria:
  - it decreases the **AIC** more than any other single regulator or has the largest log-likelihood of any of the regulators that are not already in the model.
  - the number of regulators in model is smaller than a certain predetermined number, set to 10 in our implementation.
- *Step 7*. Compute the quadratic error with respect to the expression level of the target gene of its estimated value from the parameters computed in AIC forward selection procedure (steps 5 and 6).

Figure 3. Comparative plot between the observed and the predicted values of mRNA expression levels of gene YALO31C, YBR047W, YBR137W. Examples of good estimation of the expression profile with the sigmoid pattern of regulation.



The performance of this algorithm is expressed by an order of magnitude equal to $O(nm^2)$ for the case of the beta sigmoid function and $O(n^2m^2)$ for the case of the sigmoid as regulatory function. Since for actual experimental data the number of time courses n is quite small the difference in the performance of the two algorithms comes from the fact that the search of the maximum is less costly than the computation of the statistical parameters for a data set.

Therefore, overall the algorithm is not very expensive; yet as with most optimization procedures, stepwise variable selection is a locally optimal procedure, and it may get stuck in a local maximum/minimum solution (Li and Nyholt, 2001).

RESULTS

Yeast Cell Cycle Microarray Data

We evaluated our method on a well studied data set containing gene expression measurements of the mRNA levels of 6178 S. cerevisiae ORFs at 18 time points under the α factor synchronization method from Spelmann et al., 1998. We used a candidate pool of regulators containing 216 potential regulator from http://www.csie.ntu.edu.tw/~b89x035/yeast, constructed by joining transcription factors, cell-cycle control factors and DNA-binding transcriptional regulators described in the literature (Spellman et al., 1998; Harbison et al., 2004; Chen et al., 2004). This set has been created with respect to the regulation of the cell cycle process. We analyzed the entire data set even if only about 800 genes have been identified to be involved in the cell cycle of the budding yeast (Spellman et al., 1998). There is no methodological artifact since the target genes are processed independently. The benefit is that good prediction results may lead to new hypotheses on the regulators of a particular gene. The output of our analysis is bipartite. For each gene we provide

1. the parameters of the goodness of fit: log likelihood (log L), AIC and quadratic error (QE) of the predicted mRNA levels with respect to the observed values;

Figure 4. Comparative plot between the observed and the predicted values of mRNA expression levels of gene YDRO84C, YPL143W, YPR204W. Examples of good estimation of the expression profile using beta sigmoid pattern of regulation.

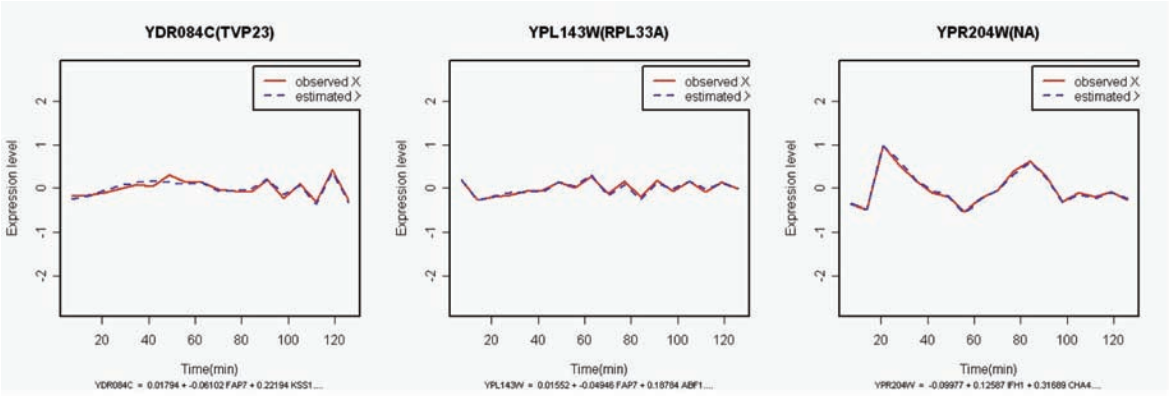


Table 1. Fitting parameters for the gene expression levels represented in the figures 3 and 4

Gene name	L	AIC	QE	Fitting
YAL031C(FUN21)	17.85	-21.71	0.08	YAL031C = -0.50836 + 0.50288 GAL4 + 0.33767 ASK10 + 0.46422
FKH2 +	22.76	-33.52	0.04	-0.20479 CRZ1 + 0.17505 DAL80 + 0.10282 HAP5
YBR047W(FMP23)	23.11	-38.21	0.07	YBR047W = -0.05932 + -0.25765 HAC1 + 0.09219 HSF1 + 0.10295 +
HAL9	18.66	-25.32	0.09	0.1377 GAL80 + 0.0687 FZF1
YBR137W(NA)	26.89	-43.78	0.04	YBR137W = 0.06551 + 0.04099 IFH1 + -0.18184 GAT3 + -0.23632 CBF1
YDR084C(TVP23)	23.09	-32.19	0.03	YDR084C = 0.01794 + -0.06102 FAP7 + 0.22194 KSS1 + -0.15343 -0.14782
GAL4 +				GCR2 + 0.09741 HIR3
YPL143W(RPL33A)				YPL143W = 0.01552 + -0.04946 FAP7 + 0.18784 ABF1 + -0.08233 -0.08542
FKH1 +				DAT1
YPR204W(NA)				YPR204W = -0.09977 + 0.12587 IFH1 + 0.31689 CHA4 + -0.22244 DAT1 +
				0.22862 HAP2 + -0.08835 DAL82 + -0.14318 IXR1

2. the corresponding regulators with their regulatory effect expressed by the local network weights; positive weights correspond to **activator** genes and negative weights correspond to **repressor** genes.

Table 1 shows an example of well fitted genes and their parameters. A linear SDE model has been analyzed for this data set by Chen et al. (2005) for a sigmoid regulatory function and by Climescu-Haulica and Quirk (2007) for a beta sigmoid function. When applied with a sigmoid regulatory function the nonlinear SDE provides better results for 43% of genes compared with the results from Chen et al., 2005. The nonlinear SDE with beta sigmoid function provides an improvement for 32% of genes compared with the results obtained from the linear model. Overall, we note that the number of well fitted genes (QE < 0.5) increased to 2365.

Figure 3 shows several examples of dynamical transcriptional patterns fitted with the nonlinear SDE method and Table 1 shows the corresponding parameters of fit.

D. Melanogaster Embryogenesis Data

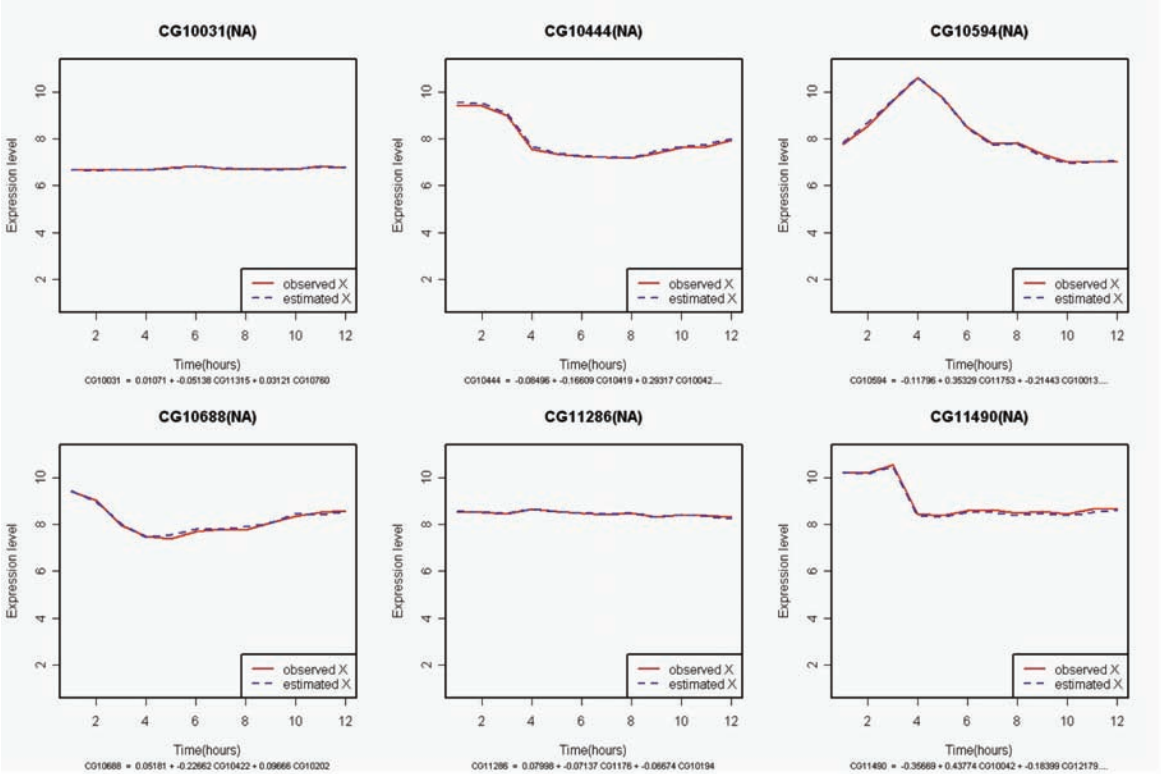
The second data set analyzed has been collected from Berkeley Drosophila Genome Project (BDGP) gene expression database (Tamancak et al., 2002) providing gene expression measurements of the mRNA levels of 3418 D. melanogaster ORFs at 12 time points using Affymetrix Gene Chip technology.

In order to choose the list of potential regulators we conduct the analysis in three groups as selected by Hooper et al., 2007:

1) maternal: genes encoding transcripts that start with a high relative transcript level, which subsequently decreases,
2) transient: genes whose transcripts levels first increase and later decrease, and
3) activated: genes encoding transcripts in expression increase only.

For each gene we provide the parameters of the goodness of fit: log likelihood (log L), AIC and quadratic error of the predicted mRNA levels with respect to the observed values (QE) the corresponding regulators with their regulatory effect expressed by the local network weights; positive weights correspond to **activator** genes and negative weights correspond to **repressor** genes.

Figure 5. Examples of good estimation of the expression profile with the nonlinear SDE method for Berkeley Drosophila Genome Project gene expression database. The shape of the expression profile almost flat makes the decision about the right regulators more difficult.



In particular, the D. melanogaster dataset contains a large subset of genes with almost flat expression profile. This fact makes the prediction of regulators more difficult, as the **forward selection** strategy could easily produce non optimal results. We tried to overcome this effect by comparing the prediction output for the sigmoid and betasigmoid models, and counting as good results the genes for which the expression level were well fitted by one and poorly fitted by the other.

For this data set we obtain a good fit (QE < 0.5) for 1534 genes from which 992 from a betasigmoid regulatory function setting. Among these genes 21 have a perfect fit (QE=0). They are listed in Table 2 which provides the fitting parameters and the regulators with their contribution.

This result is not surprising: because the expression level for the D. melanogaster do not vary too much in time the difficulty is not to obtain a very good fit but to insure that the pool of genes chosen as regulators is appropriate. This kind of validation could be done only by promoter analysis tools, binding sites, transcription factor analysis.

Table 2. Examples of D. melanogaster genes with perfect mean square estimation of the expression profile from the nonlinear SDE model

Gene name	L	AIC	QE	Fitting
CG10031(NA)	24.69	-43.39	0	CG10031 = 0.01071 + -0.05138 CG11315 + 0.03121 CG10760
CG10185(NA)	28.73	-49.46	0	CG10185 = 0.00734 + -0.03509 CG11315 + 0.02993 CG10206 + -0.01421
CG10396(NA)	28.63	-51.27	0	CG10089
CG10505(NA)	29.36	-54.71	0	CG10396 = -0.01438 + 0.04899 CG10419 + -0.02089 CG11574
CG11018(NA)	24.47	-42.94	0	CG10505 = 0.01056 + -0.02189 CG10625
CG11037(NA)	28.98	-51.96	0	CG11018 = -0.05324 + 0.06787 CG12175 + 0.03717 CG11317
CG11286(NA)	23.83	-39.66	0	CG11037 = 0.02815 + -0.03372 CG11091 + -0.02139 CG12591
CG11459(NA)	24.33	-40.65	0	CG11286 = 0.07998 + -0.07137 CG1176 + -0.06674 CG10194 + -0.03149
CG11821(NA)	26.88	-47.76	0	CG11569
CG11928(NA)	31.83	-55.67	0	CG11459 = 0.05677 + -0.06126 CG10426 + -0.03718 CG10205 + -0.02012
CG12069(NA)	26.98	-47.96	0	CG10431
CG12309(NA)	26.9	-45.81	0	CG11821 = 0.00204 + -0.03304 CG10202 + 0.02679 CG11570
CG12334(NA)	24.59	-43.17	0	CG11928 = -0.02328 + 0.03445 CG10431 + 0.03601 CG10006 + -0.01967
CG12347(NA)	29.04	-50.08	0	CG10627
CG12490(NA)	24.97	-43.95	0	CG12069 = 0.01385 + -0.05704 CG10202 + 0.02819 CG10759
CG12520(NA)	23.03	-38.07	0	CG12309 = 0.00734 + 0.05874 CG1158 + -0.0549 CG10623 + -0.01861
CG12612(NA)	24.74	-41.49	0	CG10425
CG10031(NA)	24.69	-43.39	0	CG12334 = 0.04106 + -0.04679 CG12593 + -0.03751 CG12173
CG10031(NA)	24.69	-43.39	0	CG12347 = -0.02877 + 0.03129 CG11091 + 0.03795 CG10761 + -0.01388
CG10185(NA)	28.73	-49.46	0	CG11099
CG10031(NA)	24.69	-43.39	0	CG12490 = -0.01137 + 0.08886 CG12593 + -0.068 CG10205
				CG12520 = -0.01036 + 0.0398 CG10011 + -0.043 CG12589 + 0.02219
				CG11320
				CG12612 = 0.05958 + -0.02909 CG10426 + -0.05538 CG1176 + -0.0365
				CG10428
				CG10031 = 0.01071 + -0.05138 CG11315 + 0.03121 CG10760
				CG10031 = 0.01071 + -0.05138 CG11315 + 0.03121 CG10760
				CG10185 = 0.00734 + -0.03509 CG11315 + 0.02993 CG10206 + -0.01421
				CG10089
				CG10031 = 0.01071 + -0.05138 CG11315 + 0.03121 CG10760

DISCUSSION

Gene-expression profile data is a pioneering work in network interactions and employs a variety of clustering methods in an attempt to group genes with similar patterns (Eisen at al., 1998; Tavazoie et al., 1999). Another widely-used application organized co-expressed genes by promoter sequence motif (Bussermaker et al., 2001). In recent years a variety of quantitative methods in processing expressed cell cycle yeast data have been explored: fuzzy logic approach (Woolf and Wang, 2001), the smooth surface response (SRS) algorithm (Xu et al., 2002), Bayesian inference (Li et al, 2007)), the module networks (Segal et al., 2003), etc. The fuzzy logic approach and SRS algorithm proposed the creation of a connected network and the building of a model to find triplets of **activators**, **repressors** and target genes. Such algorithms construct quite well parts of gene regulatory machinery in yeast cell cycle. The method presented here addresses this question from a stochastic and dynamic system point of view. It provides a novel study in characterizing the time series expression level based on SDE theory and kinetic interactions, as well as identifying possible regulators by proper statistical approach. This framework not only provides the regulatory relationship between regulators and target genes, but also quantifies the regulatory abilities to the specific target genes. It concerns a method of multi-regulators, but not a triplet model of activator, repressor and target gene. The model takes in account each individual gene which

better reflects the reality when compared with module network method (Segal et al., 2002) for which the same regulators are inferred for a bunch of genes. Moreover, the SDE model may find the eventual loops of the network, fact which is unaffordable for Bayesian methods, for example.

The regulation of gene expression in eukaryotes is a complex phenomenon and various particularities from one type of gene to another may occur. Hence the regulatory pattern can vary from gene to gene (Pilpel et al., 2001). This fact is revealed in our result which shows that there are genes for which we can choose the best model between the beta sigmoid and the sigmoid pattern while for other genes neither of them fits the data. Before reaching this conclusion one has to be aware about the limitation induced from the selection of the set of potential regulators since incomplete information at this level may deteriorate the results.

Computationally, the nonlinear SDE method has several advantages over other implementations. The fitted curves adequately depict differently shaped expression patterns while keeping the model parameters as few as possible. Applying more advanced results of SDEs theory, the algorithm can be implemented in more complex dynamic biological systems.

This method may be improved with respect to several aspects. First, here we were not concerned about interactions between regulators in the model of combinational regulatory functions; the real-world regulatory mechanism is more complicated than as assumed in the model. Second, the AIC **forward selection** might not be good enough for model selection. A better optimization procedure as the Cross Entropy method (Rubinstein et. al, 2004) could be applied later. In particular, the model selection is critical for a dataset which contains a large subset of gene expression with a flat profile. Third, if there are true regulators that have not been identified in literature, they are not included in the candidate pool. As a result, they would never be identified as regulators by the algorithm. The method may benefit from further discoveries related with the search for the transcription factor binding sites - pieces of DNA that serve as molecular switches to turn genes on and off – which lately made considerable progress (Shultzaberger et al., 2007 ; Segal et al., 2008).

On the other hand, our model could be used to extend methods on the identification of transcription factor cooperativity (Chang et al., 2006). These methods focus on finding transcription factor pairs while the nonlinear SDE model may reveal combination of transcription factors. The literature show that several methods on the transcription factor analysis applied on the Spellman data for the yeast cell

*Table 3. Examples of S. cerevisiae regulatory network obtained from transcription factor analysis extracted from literature. The nonlinear SDE method could improve this type of results by determining the nature (***activator/repressor***) of the transcription factors.*

Target gene	Regulators via transcription factor mechanism	Literature evidences
Swi4	Swi6, Fkh2, Ndd1, Stb1, Ste12	Tsai et al. (2005); Chang et al. (2006)
Swi6	Mbp1, Fkh2, Fkh1, Swi4, Stb1	Tsai et al. (2005); Chang et al. (2006); Ho et al. (1999)
Yap5	Msn4, Hap4, Rap1, Dat1, Hap1, Rgm1, Swi5, Gat3	Tsai et al. (2005); Chang et al. (2006); Manke et al. (2003)
Gat3	Yap5, Pdr1, Hap4, Rap1, Rgm1	Manke et al. (2003); Tsai et al. (2005); Chang et al. (2006);
Msn4	Yap5, Pdr1, Hap1, Dat1	Tsai et al. (2005); Chang et al. (2006)
Fkh1	Fkh2, Swi6, Mbp1, Mcm1	Tsai et al. (2005); Chang et al. (2006); Kumar et al. (2000)
Pdr1	Rgm1, Gat3, Msn4, Hap4, Snp1	Manke et al. (2003); Tsai et al. (2005); Chang et al. (2006)

cycle revealed a structure for the regulatory network of S. cerevisiae. We present in Table 3 an example of transcription factor cooperativity for a representative set of genes.

Overall, the results indicate that the nonlinear SDE model can capture the profile of the transcriptional regulatory throughput quite accurately. It constitutes a tool to generate hypotheses about the regulatory network structure and to be used in conjunction with qualitative bioinformatics tools.

CONCLUSION

The mathematical methodologies applied to computational biology evolve from discrete combinatorial approaches to continuous dynamical tools. This type of models follows the production of temporal gene expression data available and allows the integration of a larger number and categories of parameters. Simultaneously, more realistic models are obtained by considering the stochasticity of the molecular phenomena (Chen et al., 2005; Climescu-Haulica and Quirk, 2007).

The method described here keeps track of the temporal variation of the mRNA degradation rate from kinetic interactions and uses a nonlinear **stochastic differential equation** model to reverse engineer gene regulatory networks from time series data. We show that this model improves the prediction of target gene expression profiles of S. cerevisiae in comparison with the linear stochastic differential equation model with a sigmoid (Chen et al., 2005) and a betasigmoid regulatory pattern (Climescu-Haulica and Quirk, 2007). Applied on D. Melanogaster data base (Tomancak et al., 2002) the method generates hypotheses about regulatory relationships between genes during embryogenesis. This study shows that the nonlinear SDE framework constitutes a reliable tool for the analysis of the transcriptional regulatory networks, when completed with a validation of the identified regulators by a promoter analysis. The SDE framework has the advantage of plasticity: it may adapt to different types of data or experimental conditions when corresponding transcription rate, regulatory functions or noise models are found. Used in the form presented here or extended to more complex settings, this method may have a drastic impact for the consolidation of the knowledge about the gene regulatory networks.

FUTURE RESEARCH DIRECTIONS

The method described here is included in the „divide and impera“ logic which is the usual way of addressing the transcriptional regulatory network structure: the local connections - i.e., the strict neighborhood of one target gene - are inferred and the network is re-composed from fixing together all of the pieces. The information obtained in this way if still useful although not sufficient. This type of approach may lead to the loss of more subtle transcription/ regulation connections and communication patterns. For this reason future research directions may consider both, local and ``global'' transcriptional regulatory networks model settings. The local approach cannot be ignored as being a premise for the development of a ``global'' networking view. Some local directions of research emerging from the stochastic differential equation method as well as a global model are suggested in the following.

Local Level Study: Stochastic Differential Equation Method to Model Transcriptional Regulatory Network from Space-Time Gene Expression Data

Modern technologies allow obtaining images as gene expressions input data. For example, the detailed spatio-temporal pattern of expression of the gene are obtained by staining the *m*RNA of a gene via *in situ* hybridization (ISH) during the development of a *D. melanogaster* embryo (Tomancak et al., 2002). Automated computational approaches have been developed for the analysis of this type of data (Peng et all, 2007). Based on clustering methods these algorithms infer transcription factor binding site motifs for genes that appear to be co-regulated and automatically identify the anatomical regions that express a gene given a training set of annotations.

The SDE framework could be generalized to process this type of information and to infer transcriptional regulatory network from image input data. In the SDE equation a spatial Brownian Motion may be used as **noise** model and consequently a wavelets spectral representation technique allows the reuse of the temporal algorithms presented here.

Local Level Study: Stochastic Differential Equation Framework to Investigate the Transcriptional Regulatory Network Behavior Under Different Environmental Conditions

A big question of the epigenomics is how to infer information about the gene-environment interactions. This type of investigation brings an event driven dynamic perspective on modeling transcriptional regulatory networks with big impact in clinical research. The SDE model has the flexibility to switch between different types of regulatory functions in order to accommodate the fitting of gene expression obtained in different experimental conditions. For the case of S. *cerevisiae* some examples of experimental conditions allowing an event driven dynamic investigation are: sporulation, diauxic shift, heat and cold shock, treatment with DTT, Pheromone, and DNA-damaging agents. The objective is to analyze the dynamical changes of the network when the organism is exposed to environmental conditions, both internal and external. For the SDE method this request translates into finding the prototype of regulatory function which model best the temporal expression data for each particular condition.

From there the estimation of the level and the nature (activator/repressor) of contribution for each transcriptional regulator corresponding to a given environmental condition will follow.

Global Level Study: Transcription Clusters Gene - Gene Interactions

The big picture of the transcriptional regulatory network cannot ignore the existence of epigenetic events which seems to be directed by trans-acting factors and cis-regulatory sequences in the vicinity of the genes (Costa, 2003). The concept of “gene” seen as a “transcription cluster” in eukaryotes (Mattick, 2002) accommodates conceptually a global level network model based on “gene” - “gene” relationships mediated by intronic and exonic ncRNAs. In particular, this model might interrelate with genes signaling at the epigenetic level. It is this type of networking concept that constitutes an expected place for a system biology view, integrating results from various type of studies: genetic, molecular, biochemical, bioinformatics, etc. An interesting tool to model such type of realistic problems including heterogeneous time varying data is given by the fuzzy differential equations (Masoud et al., 2004). The nonlinear SDE method could be adapted towards a fuzzy stochastic differential equation to keep track of the system randomness.

REFERENCES

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, AC-19,* 716–723. doi:10.1109/TAC.1974.1100705

Bussemaker, H. J., Li, H., & Siggia, E. D. (2001). Regulatory element detection using correlation with expression. *Nature Genetics*, *27,* 167–171. doi:10.1038/84792

Casella, G., & Berger, R. (2001). *Statistical inference.* Belmont, CA: Duxbury Press.

Chang, Y. H., Wang, Y. C., & Chen, B. S. (2006). Identification of transcription factor cooperativity via stochastic system model. *Bioinformatics (Oxford, England)*, *22*, 2276–2282. doi:10.1093/bioinformatics/btl380

Chen, H. C., Lee, H. C., Lin, T. Y., Li, W. H., & Chen, B. S. (2004). Quantitative characterization of the transcriptional regulatory network in the yeast cell cycle. *Bioinformatics (Oxford, England)*, *20*, 1914–1927. doi:10.1093/bioinformatics/bth178

Chen, K. C., Wang, T. Y., Tseng, H. H., Huang, C. Y., & Kao, C. Y. (2005). A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae. *Bioinformatics (Oxford, England)*, *21*, 2883–2890. doi:10.1093/bioinformatics/bti415

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., & Wodicka, L. (1998). A genome wide transcriptional analysis of the mitotic cell cycle. *Molecular Biology of the Cell*, *2*, 65–73.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., & Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science*, *282*, 699–705. doi:10.1126/science.282.5389.699

Chung, K. L., & Williams, R. J. (1990). *Introduction to stochastic integration*. Boston: Birkhäuser.

Clamp, M., Fry, B., Kamal, M., Xie, X., Cuff, J., & Lin, M. F. (2007). Distinguishing protein-coding and noncoding genes in the human genome. *Proceedings of the National Academy of Sciences of the United States of America*, *104*, 19428–19433. doi:10.1073/pnas.0709013104

Claverie, J. M. (2005). Fewer genes, more noncoding RNA. *Science*, *309*, 5740. doi:10.1126/science.1116800

Climescu-Haulica, A., & Quirk, M. D. (2007). A stochastic differential equation model for transcriptional regulatory networks. *BMC Bioinformatics*, *8*(Suppl 5), S4. doi:10.1186/1471-2105-8-S5-S4

Costa, F. (2007). Noncoding RNAs: New players in eukaryotic biology. *Gene, 357*(2), 83–94. doi:10.1016/j.gene.2005.06.019

Eisen, M. B., Spelmman, P. T., Brown, P. O., & Botstein, O. (1998). Cluster analysis and display of genomewide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(25), 14863–14868. doi:10.1073/pnas.95.25.14863

Guet, C. C., Elovitz, M. B., Hsing, W., & Leibler, S. (2002). Combinatorial synthetis of genetic networks. *Science*, *296*, 1466–1470. doi:10.1126/science.1067407

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., & Danford, T. W. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, *431*, 99–104. doi:10.1038/nature02800

He, H., Cai, L., Skogerbø, G., Deng, W., Liu, T., & Zhu, X. (2006). Profiling Caenorhabditis elegans noncoding RNA expression with a combined microarray. *Nucleic Acids Research*, *34*(10), 2976–2983. doi:10.1093/nar/gkl371

Ho, Y. H., Constanzo, M., & Moore, L. (1999). Regulation of transcription at the Saccharomyces cerevisiae start transition by Stb1, at a Swi6-binding protein. *Molecular and Cellular Biology*, *19*, 5267–5278.

Hooper, S. D., Boue, S., Krause, R., Jensen, L. J., Mason, C. E., & Ghanim, M. (2007). Identification of tightly regulated groups of genes during Drosophila melanogaster embryogenesis. *Molecular Systems Biology*, *3*, 72. doi:10.1038/msb4100112

Iyer, V. R., Horak, C. E., Scafe, C. S., Botstein, D., Snyder, M., & Brown, P. O. (2001). Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, *409*, 533–538. doi:10.1038/35054095

Kaern, M., Elston, T. C., Blake, W. J., & Collins, J. J. (2005). Stochasticity in gene expression: From theory to phenotypes. *Nature Reviews. Genetics*, *6*, 451–464. doi:10.1038/nrg1615

Karatzas, I., & Shreve, S. E. (1991). *Brownian motion and stochastic calculus*. New York: Springer-Verlag.

Kumar, R., Reynolds, D. M., Shevchenko, A., Goldstone, S. D., & Dalton, S. (2000). Forkhead transcription factor Fkh1p and Fkh2p collaborate with Mcm1p to control transcription required for M-phase. *Current Biology*, *10*, 896–906. doi:10.1016/S0960-9822(00)00618-7

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., & Gerber, G. K. (2002). Transcriptional regulatory network in Saccharomyces cerevisie. *Science*, *298*, 799–804. doi:10.1126/science.1075090

Li, F., Long, T., Lu, Y., Ouyang, T., & Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, *101*, 4781–4796. doi:10.1073/pnas.0305937101

Li, P., Zhang, C., Perkins, E. J., Gong, P., & Deng, P. (2007). Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks. *BMC Bioinformatics*, *8*(Suppl 7), S13. doi:10.1186/1471-2105-8-S7-S13

Li, W., & Nyholt, D. R. (2001). Marker selection by Akaike information criterion and Bayesian information criterion. *Genetic Epidemiology*, *21*(Suppl. 1), S272–S277.

Lu, J., Ruhf, M. L., Perrimon, N., & Leder, P. (2007). A genome-wide RNA interference screen identifies putative chromatin regulators essential for E2F repression. *The Proceedings of the National Academy of Sciences Online (US)*, *104*(22), 9381–9386. doi:10.1073/pnas.0610279104

Manke, T., Bringos, R., & Virigron, M. (2005). Correlating protein-DNA and protein-protein interactions networks. *Journal of Molecular Biology*, *333*, 7585.

Masoud, N., Zadeh, L., & Korotkikh, V. (Eds.). (2004). *Fuzzy partial differential equations and relational equations*. Springer.

Mattick, J. (2003). Introns and noncoding RNAs: The hidden layer of Eukariotic complexity. In J. Barciszewski (Ed.), *Noncoding RNAs.* Kluwer Academic.

Novikov, E., & Barillot, E. (2008). Regulatory network reconstruction using an integral additive model with flexible kernel functions. *BMC Systems Biology*, *2*(8).

Peng, H., Long, F., Zhou, J., Leung, G., Eisen, M., & Myers, E. (2007). Automatic image analysis for gene expression patterns of fly embryos. *BMC Cell Biology*, *8*(Suppl 1), S7. doi:10.1186/1471-2121-8-S1-S7

Pilpel, Y., Sudarsanam, P., & Church, G. M. (2001). Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, *29*, 153–159. doi:10.1038/ng724

R Development Core Team. (2006). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing.

Ren, B., Robert, F., Wyrick, J., Aparicio, O., Jennings, E., & Simon, I. (2000). Genomewide location and function of DNA binding proteins. *Science*, *290*, 2306–2309. doi:10.1126/science.290.5500.2306

Rubinstein, R. Y., & Kroese, D. P. (2004). *The cross entropy method*. Springer Verlag.

Sayyed-Ahmad, A., Tuncay, K., & Peter, J. (2007). Transcriptional regulatory network refinement and quantification through kinetic modeling, gene expression microarray data, and information theory. *BMC Bioinformatics*, *8*, 20. doi:10.1186/1471-2105-8-20

Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U., & Gaul, U. (2008). Predicting expression patterns from regulatory sequence in Drosophila segmentation. *Nature*, *451*(7178), 535–540. doi:10.1038/nature06496

Shultzaberger, R. K., Chiang, D. Y., Moses, A. M., & Eisen, M. B. (2007). Determining physical constraints in transcriptional initiation complexes using DNA sequence analysis. *PLoS ONE*, *2*(11). doi:10.1371/journal.pone.0001199

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V., Anders, K., & Eisen, M. B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by micro-array hybridization. *Molecular Biology of the Cell*, *9*, 3273–3297.

Tan, W.-Y. (2002). *Stochastic models with applications to genetics, cancers, AIDS, and other biomedical systems*. World Scientific Publishing Co. Ltd.

Tavazoie, S., Hugues, J. D., Campell, M. J., Cho, R. J., & Church, G. M. (1999). Systematic determination of genetic network architecture. *Nature Genetics*, *22*, 281–285. doi:10.1038/10343

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S. Q., & Lewis, S. E. (2002). Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biology*, *3*(12). doi:10.1186/gb-2002-3-12-research0088

Tsai, H. K., Lu, H. H., & Li, W. H. (2005). Statistical methods for identifying yeast cell cycle transcription factors. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 13532–13537. doi:10.1073/pnas.0505874102

Vu, T., & Vohradsky, J. (2007). Nonlinear differential equation model for quantification of transcriptional regulation applied to microarray data of Saccharomyces cerevisiae. *Nucleic Acids Research*, *35*(1), 279–287. doi:10.1093/nar/gkl1001

Weisberg, S. (1985). *Applied linear regression.* New York: John Wiley.

Woolf, P. J., & Wang, Y. (2000). A fuzzy logic approach to analyzing gene expression data. *Physiological Genomics*, *3*, 9–15.

Xu, H., Wu, P., Wu, C. F., Tidwell, C., & Wang, Y. (2002). A smooth response surface algorithm for constructing a gene regulatory network. *Physiological Genomics*, *11*, 11–20.

ADDITIONAL READING

Basso, K., Margolin, A. A., Stolovitzky, G., Klein, U., Dalla-Favera, R., & Califano, A. (2005). Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, *37*, 382–390. doi:10.1038/ng1532

Battle, A., Segal, E., & Koller, D. (2005). Probabilistic discovery of overlapping cellular processes and their regulation. *Journal of Computational Biology*, *12*(7), 909–927. doi:10.1089/cmb.2005.12.909

Beer, M. A., & Tavazoie, S. (2004). Predicting gene expression from sequence. *Cell*, *117*, 185–198. doi:10.1016/S0092-8674(04)00304-6

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*, 67–103. doi:10.1089/10665270252833208

Dekker, J. (2008). Gene regulation in the third dimension. *Science*, *319*(5871), 1793–1794. doi:10.1126/science.1152850

Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *301*, 102–105. doi:10.1126/science.1081900

Ghosh, P., Ghosh, S., Basu, K., & Das, S. K. (2007). A Markov model based analysis of stochastic biochemical systems. *Computational Systems Bioinformatics / Life Sciences Society. Computational Systems Bioinformatics Conference*, *6*, 121–132.

Heron, E. A., Finkenstadt, B., & Rand, D. A. (2007). Bayesian inference for dynamic transcriptional regulation: The Hes1 system as a case study. *Bioinformatics (Oxford, England)*, *23*(19), 2596–2603. doi:10.1093/bioinformatics/btm367

Hirose, O., Yoshida, R., Imoto, S., Yamaguchi, R., Higuchi, T., & Charnock-Jones, D. S. (2008). Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models. *Bioinformatics (Oxford, England)*, *24*(7). doi:10.1093/bioinformatics/btm639

Hobert, O. (2008). Gene regulation by transcription factors and microRNAs. *Science*, *319*(5871), 1785–1786. doi:10.1126/science.1151651

Makayev, E. V., & Maniatis, T. (2008). Multilevel regulation of gene expression by microRNAs. *Science*, *319*(5871), 1789–1790. doi:10.1126/science.1152326

Margolin, A. A., & Califano, A. (2007). Theory and limitations of genetic network inference from microarray data. *Annals of the New York Academy of Sciences*, *1115*, 51–72. doi:10.1196/annals.1407.019

Moses, A. M., Pollard, D. A., Nix, D. A., Iyer, V. N., Li, X. Y., Biggin, M. D., & Eisen, M. B. (2006). Large-scale turnover of functional transcription factor binding sites in Drosophila. *PLoS Computational Biology*, *2*(10). doi:10.1371/journal.pcbi.0020130

Quach, M., Brunel, N., & D'Alche-Buc, F. (2007). Estimating parameters and hidden variables in nonlinear state-space models based on ODEs for biological networks inference. *Bioinformatics (Oxford, England)*, *23*(23), 3209–3216. doi:10.1093/bioinformatics/btm510

Rosenfeld, S. (2007). Stochastic cooperativity in nonlinear dynamics of genetic regulatory networks. *Mathematical Biosciences*, *210*(1), 121–142. doi:10.1016/j.mbs.2007.05.006

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., & Friedman, N. (2003). Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, *34*(2), 166–176. doi:10.1038/ng1165

Shieh, G. S., Chen, C. M., Yu, C. Y., Huang, J., Wang, W. F., & Lo, Y. C. (2008). Inferring transcriptional compensation interactions in yeast via stepwise structure equation modeling. *BMC Bioinformatics*, *9*(134).

Sinha, S., Adler, A. S., Field, Y., Chang, A. Y., & Segal, E. (2008). Systematic functional characterization of cis-regulatory motifs in human core promoters. *Genome Research*, *18*(3), 477–488. doi:10.1101/gr.6828808

Soranzo, N., Bianconi, G., & Altafini, C. (2007). Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: synthetic versus real data. *Bioinformatics (Oxford, England)*, *23*(13), 1640–1647. doi:10.1093/bioinformatics/btm163

Storey, J. D., Akey, J. M., & Kruglyak, L. (2005). Multiple locus linkage analysis of genomewide expression in yeast. *PLoS Biology*, *3*(8). doi:10.1371/journal.pbio.0030267

Stuard, J. M., Segal, E., Koller, D., & Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, *302*(5643), 249–255. doi:10.1126/science.1087447

Tian, T., & Burrage, K. (2006). Stochastic models for regulatory networks of the genetic toggle switch. *The Proceedings of the National Academy of Sciences Online (US)*, *103*(22), 8372–8377. doi:10.1073/pnas.0507818103

Tian, T., Xu, S., Gao, J., & Burrage, K. (2007). Simulated maximum likelihood method for estimating kinetic rates in gene expression. *Bioinformatics (Oxford, England)*, *23*(1), 84–91. doi:10.1093/bioinformatics/btl552

Tsai, K. Y., & Wang, F. S. (2005). Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics (Oxford, England), 21*, 1180–1188. doi:10.1093/bioinformatics/bti099

von Seggern, D. (1993). *CRC standard curves and surfaces*. Boca Raton, FL: CRC Press.

KEY TERMS AND DEFINITIONS

Stochastic Calculus: The framework which permits to define rigorous theory of integration for stochastic processes.

Transcription Rate: The variation in time of the mRNA production for a given gene.

Regulation Function: The quantitative time dependent form on which the regulators interfere on the mRNA target gene production.

Local Regulatory Network: The set of regulators corresponding to a single target gene.

Model Selection: The procedure from which a statistical model is selected from a set of potential models, given the data; usually that corresponds to the choice of a set of parameters.

Learning Relationship: Plastically link between two entities (in our case genes); it adapts during the time with respect to various stimulus.

Goodness of Fit: The measure for how well a statistical model fits a set of observations.

Chapter 10
Modelling Gene Regulatory Networks Using Computational Intelligence Techniques

Ramesh Ram
Monash University, Australia

Madhu Chetty
Monash University, Australia

ABSTRACT

This chapter presents modelling gene regulatory networks (GRNs) using probabilistic causal model and the guided genetic algorithm. The problem of modelling is explained from both a biological and computational perspective. Further, a comprehensive methodology for developing a GRN model is presented where the application of computation intelligence (CI) techniques can be seen to be significantly important in each phase of modelling. An illustrative example of the causal model for GRN modelling is also included and applied to model the yeast cell cycle dataset. The results obtained are compared for providing biological relevance to the findings which thereby underpins the CI based modelling techniques.

INTRODUCTION

Biological processes and systems can be abstracted as multi-layered networks interacting with each other to create a complete biological system. Understanding the interactions of genes plays a vital role in the analysis of complex biological systems. The system level view of gene functions provided by gene regulatory networks (GRNs) is of tremendous importance in uncovering the underlying biological process of living organisms, providing new ideas for treating complex diseases, and for designing of new drugs. This chapter presents computational intelligence applications generally to bioinformatics problems and specifically to model networks of genetic regulation or gene regulatory networks (de Jong, 2002) (Someren et al, 2002) (Brazhnik et al, 2002) using gene expression data.

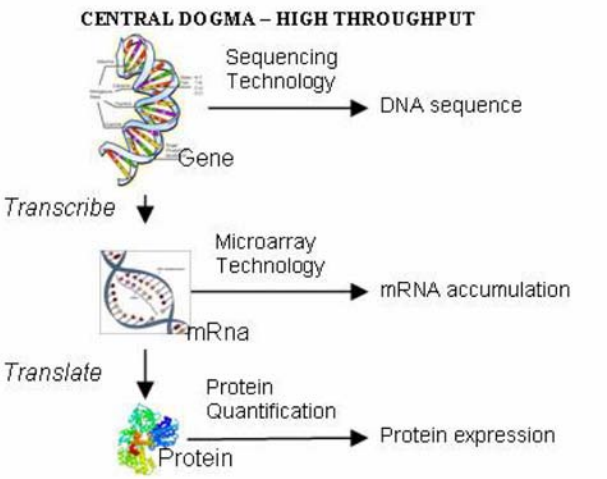
DOI: 10.4018/978-1-60566-685-3.ch010

Living beings are endowed with highly complex information storage and processing systems that are regulated in many different ways. The control of the body is carried out by large networks of regulatory genes, otherwise known as Gene Regulatory Networks (GRN). GRNs are collections of gene-gene regulatory relations in a genome that display relationships between gene activities. Increases in complexity of organisms do not bring an increase in the number of genes in the genome. For example, humans are believed to have about 20,000-25,000 genes (considerably lower than the original estimate), which is not dissimilar to the gene content for less complex organisms, such as the worm *Caenorhabditis elegans*, while a simple organism such as *Drosophila melanogaster*, also known as the fruit fly has about 14,000 genes. Therefore, the complexity may be due to a phenomena such as regulation of expression of genes in both temporal and spatial manners. A prerequisite for cellular behaviour is that the correct genes are expressed in the correct cell over correct time intervals and at correct expression levels. Regulatory networks specify how this gene expression or cellular behaviour is controlled. Over the past two decades, advances in molecular biology, DNA sequencing, and other high-throughput methods have resulted in a vast amount of bioinformatics data as shown in Fig1, including pathway information such as metabolic and regulatory pathways for various organisms. The rapid increase of this data for various organisms offers the possibility to perform analyses for single organisms (intra-species) as well as across different organisms (inter-species). However, the sheer quantity of data generated has exceeded the capacity of a researcher to extract useful information using traditional data analysis techniques. Since the high-throughput data acquisition technology for gene expression measurement known as biological Microarray technology emerged in the late 1990s, application of data mining, machine learning, and computational intelligence techniques to microarray data analysis has drawn attention of the bioinformatics community. Along with this, a significant amount of attention has been focused on modelling genetic regulatory networks from gene expression data. Microarrays allow the monitoring of expression levels of thousands of genes simultaneously and the data provide the basis to discover gene regulation networks, life evolution, and other important bio-problems. However, gene expression microarray data is characterized as massive, heterogeneous (with high dimensions), net character in nature, irregular sampling rate, having measurement errors (leading to noisy data) Hence, its analysis is beyond the ability of traditional analysis methods and decision supporting technologies. Due to the nature of data, previous deterministic models are not able to capture the time-varying dependencies between the different genes in the gene regulatory network. Researchers dealing with gene microarray data are faced with daunting quantities of data in which lies important hidden information, including transcription factor activity profiles.

There are two main objectives for modelling genetic regulatory networks (GRNs). The first is to be able to infer the regulatory network from data, in order to be able to understand the mechanisms behind them. The second is to be able to use the inferred networks in order to be able to predict the behaviour of actual networks for the purpose of diagnosing diseases and the development of drug targets and treatments. Needless to say, the system level view provided by gene networks of gene functions is of tremendous importance in uncovering the underlying biological process of living organisms, providing new ideas for treating complex diseases, and the design of new drugs. Inevitably, its research has become important in the recent times and the extensive research on this topic is timely.

Computational Intelligence (CI) is an area of computer science and engineering that deals with mimicking the intelligence observed from the natural behaviour in, for example, biology, insect societies (ants), neural sciences, immune systems. System Biology (Kitano, 2001), a sub group of computational biology, is a new and developing field of research in bioinformatics which deals with analysing large scale biological systems. It is an interdisciplinary research area combining the fields of (cell) biology and

Figure 1. High throughput technologies



systems theory. In the field of systems biology, there has been considerable discussion of algorithm-based versus literature-based modelling approaches. The algorithm-based approaches utilize data generated solely by novel high-throughput techniques at the gene, protein, or metabolite level, while excluding the data generated using more traditional approaches. Alternatively, literature-based approaches attempt to incorporate data from numerous sources and at various levels of organization, but they may miss important novel discoveries obtained from the integration of the broad gene or protein expression technologies. Both these approaches can be used to model gene regulatory networks. However, due to the heterogenous data sources, the literature based modelling leads to an integration nightmare. On the other hand, algorithm based approaches are successful where the resulting inferred network of regulatory processes helps researchers form new hypotheses about the behaviour of biological systems and assist with the design of further experiments (de Jong, 2002) (Friedman, 2000) (Friedman, 2004). This not only cut costs on biological experiments performed but also expedites the discovery process.

It is anticipated that modelling technologies will find clinical applications and provide great opportunities to deal with human diseases such as cancer. Computational methods for estimating gene networks can be applied to searching for drug target genes. Analysis of gene networks has different applications, e.g. in the process of target identification, drug design, and in the search for causes of genetic diseases. In basic research, these networks can be used for comparison of metabolic processes of different organisms. For example, the information on the metabolism of one organism can be used to understand the newly sequenced genome of another organism (Sirava et al, 2002).

The rest of the chapter is structured as follows: Section 2, the problem of gene regulatory network modelling is explained along with discussion on the related works. Section 3 elaborates on the different stages involved in computational modelling of gene regulatory networks and their scope for computational intelligence application at every stage. Since the details about various modelling techniques such as Boolean, Bayesian models, etc. have already been extensively reviewed in the literature, these are not covered in this chapter. Section 4 describes an illustrative experiment performed on a yeast cell cycle

gene regulatory network. Finally, section 5 provides concluding remarks on the nature of the problem and the solutions presented.

GRN MODELLING PROBLEM

In cells of living organisms, there are thousands of genes that are interacting with each other at any given time to accomplish complicated biological tasks. Genes code for proteins that are essential for development and functioning of organisms. The central dogma of molecular biology (Fig.1) outlines the flow of information contained within the DNA (gene) into messenger RNA (mRNA) molecules which, in turn, are translated into proteins. The amount of protein produced from a gene is called its expression level. The protein produced either perform functions of cells or control and regulate the flow of information of other genes in what is known as transcription factors (TFs). Since TFs are proteins, they are products of genes themselves. This implies that genes regulate the expressions of other genes. As a matter of fact, much of the complex behaviour of a cell can be explained through this concerted activity of genes. This activity is typically represented as a network of interacting genes which is called gene regulatory network (GRN).

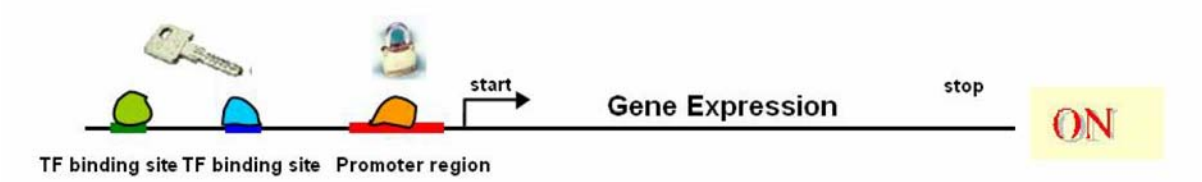
A GRN consists of genes as its nodes and the arcs in the network contributing to transcription regulations. Inferring and modelling of this network is significant for finding answers to three important questions: (1) how the cell knows which genes to transcribe into RNA and translate into protein, (2) why cells have different properties, i.e., blood cells, skin cells, liver cells, etc., when all cells in an organism contain the same DNA and (3) how a gene expression programs, or a cell-cell signal,is controlled by regulators across the genome. These regulatory networks may also contain feedback loops, non-linear interactions which can be very complex. The regulators often act in a concerted manner with regulatory agents such as transcription factors taking part in various combinations, each of which may have a specific effect, and may be present at specific developmental stages in specific cells or tissues. In order to successfully model a GRN, it is essential to have a clear and an in-depth understanding of the GRN modelling.

For a better understanding the problem of modelling a gene regulatory network, the gene regulation process is explained here with a key and lock metaphor. Fundamentally, a key is a device which is used to open/close a lock. It usually consists of a specially-shaped piece of flat metal, with teeth and/or milled grooves which fit the shape of the lock and can open the lock correctly by (usually) being rotated in the lock housing. In reference to gene regulation (Fig. 2), a particular gene of interest can be thought of as a lock. The promoter of a gene is the element (key) to unlock gene expression or the lock housing. The transcription factors (TFs) are the teeth found in the key. There can be one transcription factor (a tooth) or a combination of transcription factors (teeth) involved in forming a key. The key to unlock gene expression is thus a protein complex. When the complete shape of the key binds the lock housing (gene) correctly, the transcriptase binds to the promoter region of the gene, hence causing the gene to produce its protein, or in other words, opening the lock. When the key is removed from the lock housing, the gene goes inactive, or in other words the lock is closed. Essentially, the state of a man-made lock, which can be opened only by a correct key, can have only two discrete values: either open or close. But in contrast, the amount of protein produced when a gene is opened is determined by the expression level of the gene which is continuous rather than discrete. The expression level of the gene is dependent not only on the expression level of the genes, which are responsible for producing the proteins that combine

to form the key (TF protein complex), but also on the environmental factors like temperature and pressure. If the influence of environmental factors is ignored, the task of network to be modelled is reduced to basically determine a set of locks and keys with continuous states. At first, this might look like a simple problem. However, the whole picture is not yet complete. There can be several different keys that can unlock the same promoter. Furthermore, genes can themselves have alternative promoters (lock housings) at various instances. In fact, a gene can have alternative promoters and alternative transcripts. This leads to modelling a complex combinatorial regulation involving multiple transcripts and multiple promoters and several combinations of the two. The man-made keys and locks are static as they do not vary with time while the gene regulation processes are time varying and work differently at different times. . The key that was used to unlock a gene during childhood may not be the key that will be used during adult hood. For example, a boy who does not have any beard during childhood finds it growing during adulthood. Another aspect of GRNs is the transient gene interaction, i.e., the genes interact with each other only for a short period of time. Transcriptional regulation is the first step in the regulation of gene expression. There is still one more important information which remained unexplained and which would be very helpful to interpret and to understand the function of GRNs that is not present in the lock and key metaphor. It is related to the strength of regulatory interactions between gene and its transcription factor complex. The basic idea of modelling a regulatory strength value is to find a relation on how strongly a regulation is up- or down- regulated compared to the completely non-inhibited or a non-activated state. This clearly indicates that activators and inhibitors can either act independently during a regulation or there can be a combined influence of activators and inhibitors. For example, for a gene with two regulators, when the regulation happens, they both can be activators or both can be inhibitors, or the first can be an activator and other can be an inhibitor. Further, both can be non-inhibited, both can be non-activated, and so on. Many complex interactions such as these on the molecular scale have been described (McAdams, 1997) (Spellman, 1998). They rely on presence of specific factors that can either enhance or inhibit the expression of certain genes. McAdams and Arkin (McAdams, 1997) point out that the time interval between switching on of the first promoter and its effect on the next promoter can vary widely across otherwise identical cells, as a result of various stochastic processes occurring within the cell. The reasons put forward for assuming the stochastic nature of processes are, e.g. degradation of gene products, spatial collision necessary before a reagent can exert its influence, and reversible reaction equations. For a cell to result in a less noisy output, it is necessary to produce more frequent transcripts with fewer proteins per transcript, which is related to a higher energy cost.

The dynamic time varying gene network incorporates feedback loops (Thomas, 1995) (Tyson and Othmer, 1978). Feedback plays an important role in the control of biological systems. The feedback loops may be denoted as ``positive'' or ``negative'', indicating the ultimate influence of one of the nodes in the loop on itself. Oscillators (genes with positive feedback) are important components of biological systems. Biochemical networks that exhibit oscillatory behaviour are used at the molecular level for essential time-keeping in the cell. In many cases, these networks involve transcriptional circuits that are intrinsically noisy. Researchers (Thieffry and Thomas, 1995) have examined how the nature of possible feedback loops (or equilibria) present in genetic networks depends on the properties of these networks. It is also shown (Wolf and Eeckman, 1998) that dynamic system behaviour, stability of equilibria and their bifurcation potential are largely determined from regulatory feedback loops. Molecular mechanisms underlying programs are now under intensive investigation. Recent studies strongly suggest that triggering of cell proliferation, differentiation and apoptosis (cell death) depend on the cell cycle feedback

Figure 2. Gene regulation



control. For example, the tumour suppressor protein p53 plays a major role in modulating cellular functions such as DNA repair, cell cycle arrest, and apoptosis.

The underlying network of keys and locks (GRN) can be thought of as a highly secure network protected against viruses. Any disruption to the network can lead to serious medical conditions such as cancer. The security is maintained by opening and closing of correct genes (locks) over correct time intervals to correct expression levels using the correct keys (TFs). Beyond this, a cancerous cell, when it expresses within a complex secure network, acts like a intruder which results in either opening/closing of wrong genes (locks), gene loss, additional gene, wrong timing, over expression levels and so on. During an abnormal interaction due to intrusion, genes produce keys that disrupt the functioning of the normal gene (protein). Since the network is highly connected and acts in a concerted manner having a cause and effect interactions, disrupting one gene can trigger a disturbance to a cluster of related genes which then can spread out. Fortunately, since the gene regulatory networks are sparse (Savageau, 1998), the network responsible for an activity, say the functioning of lungs, may only be weakly (or not at all) be connected with another network, say a network that supports the kidney function. So the cancer can restricted only to the affected area and does not spread to other parts of the body (although it is the same DNA). Modelling such a complex time varying sparse cyclic network is both daunting and exciting.

Early studies on gene regulation have focused primarily on the group properties of clustered genes or in making use of data mining methods for understanding this dynamic process. Subsequently, researchers began to realize that integrative approaches, e.g. computational intelligence based approaches, are needed to model different modes of the complex combinatorial dynamics of gene networks. In a typical microarray dataset, the number of observations n (with an order of tens) is substantially smaller than the number of variables p (with an order of hundreds or even thousands). Moreover, the data is characterized as massive, heterogeneous, high-dimensional and net character in nature. The small number of samples leads to a temporal aggregation bias (Bay et al, 2004) while the measurement errors lead to noisy data and these limit the ability of traditional analysis methods. Discretized data has been frequently used for the sake of simplicity (Friedman, 2000). Although multiple regression method (D'haseleer, 2000) identified correlation between gene expression levels, it was unable to determine if genes are linked directly or indirectly through other genes thus preventing direct application for structure learning from graphical models.

Many important methods presented for inference of gene networks have focused on statistical methods, such as, Bayesian networks (Friedman, 2000), dynamic Bayesian networks (Murphy et al, 1999), relevance networks (Butte and Kohane, 2000) and graphical models (de la Fuente et al, 2004). Graphical models have emerged as powerful tools for learning, description and manipulation of conditional independencies among the genes. Representations by directed acyclic graph (DAG) include influence diagrams and Bayesian networks which allow circumventing the problems associated with inferring networks

with feedbacks. Any model of the regulatory systems cannot simply be limited to describing assembly of genes and proteins and their interconnections but it should also be able to provide explanation of the underlying interactions. Furthermore, the large amount of data should not pose adverse effects on the stability of the model. Techniques for evaluating GRN models are explained in section 3.2. However, before these are discussed,, we explain the computational modelling of GRN in the next section.

3. COMPUTATIONAL MODELLING OF GRN

Computational modelling and simulation techniques handle the complexity of modelling GRN explained in Section 2 and help understand different relationships in the network. Biochemists often conduct experiments in-vitro (in the test tube) and in-vivo (in living organisms) in order to explore observable behaviours and understand the dynamics of many gene regulation processes. However, an understanding of their dynamics is hard to obtain because most pathways of interest involve components acting simultaneously in a concerted manner. On the other hand, in-silico (in the computer) modelling techniques enable researchers to study networks in a very flexible, cheap and fast way compared to the *in vitro* and *in vivo* experiments. Computational modelling can be defined as a mathematical description of a process that has generated the observed data. In the context of gene regulatory network modelling, the pattern of interactions is described using a network structure and the observed data is mainly the microarray gene expression data. In terms of the network structure, if there are N genes that are involved in the network, there are potentially O(NxN) pairs of regulatory relations to be explored before we can derive a valid network model. In a microarray, expression levels on thousands of genes are simultaneously gathered under several different conditions. These microarray data provide abundant information on molecular interactions genome wide and thus are good for uncovering gene networks. The gene expression data is basically classified as time-series and steady-state data. In steady-state experiments, only one snapshot of gene expression under different experimental conditions (e.g. temperature) is taken, while in time series experiments, a series of snapshots are taken at fixed time intervals keeping the experimental conditions constant. Even though the microarray technology provides valuable data for the construction of gene networks, many difficulties and challenges arise at the same time.

Based on the technology used, there are two types of microarrays. The complementary DNA (cDNA) microarrays allow the measurement of gene expression levels usually based on the colour strength ratios of the two dyes (red and green). The second method uses DNA chips, also called Affymetrix Gene Chips. In this method, an array of oligonucleotide or peptide nucleic acid (PNA) probes is synthesized either on-chip or by conventional synthesis followed by on-chip immobilization. The array is exposed to labelled sample DNA, hybridized, and the identity/abundance of complementary sequences is determined. Unlike the cDNA microarray, Affymetrix use only one sample during hybridization and the colour strength of the dye reflects the relative level of mRNA accumulation. Even though the design and manufacture of Affymetrix chips is more complex than cDNA microarray, Affymetrix continues to be the market leader.

Limitations of microarray technology include:

1. Number of measurements (arrays) is very limited compared to the large number of objects (genes) known as dimensionality problem
2. Data may be under sampled

3. Fast changing information may not be captured
4. Gene expression measurements are noisy, due to variations among different individuals, low quantities of some RNAs and measurement errors
5. Measured expression values have a highly asymmetric distribution. Large variations in expression values can lead to inferring spurious causal relationships. Log transformation is one major preprocessing step to gene expression data as the distribution of data is approximated as symmetric and normal

The model construction implies abstraction and simplification of the real life system. The model can be characterised using representation and learning ability. Representation specifies all that which we can model i.e. how abstract/detailed our model of the underlying system is or can we make predictions that *explain* the data. Learning ability specifies what we can learn from the model. It provides answers to questions such as: Which aspects of the system can we identify? How effectively can we search the model space? An *in silico* modelling of gene regulatory network first involves construction of the network structure phase (Representation) followed by *validation* of the network structure using the data (Learning). In addition, there is also the optional phase of searching the space of models. Clearly, each of these three stages involves unique applications of appropriate computational intelligence techniques.

3.1. Network Structure Construction

A gene network structure can be defined using network measures. There are two network measures: Degree (or Connectivity) of a node, k, is the number of links (edges) this node has and the Degree Distribution, P(k), is the probability that a selected node has exactly k links. As shown in Fig. 3 below, there are three broad types of networks based on their connectivity. These are: i) Random Network ii) Scale-Free Network and iii) Hierarchical Network. These are briefly discussed.

3.1.1. Random Networks

The classic model of random networks (Fig. 3) is based on a given number of vertices or links. The model was introduced by (Gilbert, 1959). Each pair of nodes is connected by *N* vertices with probability *p,* creating a graph with approximately *p N*(N-1)/2* randomly placed links. The connectivity degree follows a Poisson distribution, i.e., nodes that deviate from the average are rare and decrease exponentially. In random networks, the clustering coefficient is independent of a node's degree of connectivity. The mean shortest path is given as $l \sim \log(N)$ indicating that most nodes are connected by a short path (Small World model) (Cancho and Sole, 2001). A network with large Clustering Coefficient and small Average Path Length is called a *Small World Network* model. While random graphs have been studied for a long time, these standard models appear to be inappropriate because they do not share the characteristics observed in complex systems like GRNs. A plethora of new models have therefore been proposed, but many of them are variations of the small-world model or the preferential attachment model (Wang and Chen, 2004). Preferential attachment means new nodes tend to connect to nodes with large degree of connectivity. For evaluation of network models and algorithms (learning phase), it is important to be able to quickly generate graphs according to the requirement. Software network generators such as BRITE, GT-ITM, JUNG, or LEDA (Batagelj and Brandes, 2005) perform reasonably well for analysis

Figure 3. Network types

of tens of thousands of nodes.. Hybrid evolutionary computation and many other algorithms are also being proposed attempting to solve the problem.

3.1.2. Scale-Free Networks

The degree distributions of complex gene regulatory networks do not always follow a Poisson distribution like random graphs, but might follow a power law distribution (Barabási and Albert, 2002). According to the power law distribution, the behaviour of a network system is controlled by a few important nodes, that is a majority of nodes have only a few connections, while some special nodes connect to many other nodes forming a hub, i.e., most nodes are poorly connected, while a few are highly connected (hub). This type of network is called a scale-free network (Fig.3). With the same size and the same average degree, a scale-free network has a small average path length and large clustering coefficient. The degree distribution approximates a power law: $P(k) \sim k^{-\gamma}$, where γ is the degree exponent (straight line in a Log-Log plot). The smaller the γ, the more important the role of the hubs is. Most biological networks have $2 < \gamma < 3$. For $\gamma > 3$, hubs become irrelevant and the network behaves like a random network. The mean shortest path length is proportional to log(log(N)) (i.e. much shorter than the Small World model). One feasible approach (Barabási and Albert, 2002) shows a way to construct the scale-free network. There are two key features: growth and preferential attachment. Growth of the network scale indicates that the networks is not fixed at the original scale, but grows up dynamically and develops constantly from small to middle scale, and then to large scale. The nodes in these networks preferentially attach to each other during the growth process. A newly-added node tends to preferentially link these clustering nodes with the degree of distribution following the Power Law. There is uncertainty in the use of parameters such as degree of distribution, clustering coefficient, path length, etc. and artificial intelligence based approaches are now being proposed to construct scale free networks.

3.1.3. Hierarchical Network

Small-world and scale-free models are basic characters of many gene network in real world. However, many complex organisms have inherent modular structure. To accommodate modularity, clusters combine in an iterative manner, generating a hierarchical network. The hierarchical network model (Fig. 3) integrates a scale-free topology with an inherent modular structure by generating a network that has a power-law degree of distribution with the degree exponent $\gamma = 1 + \ln 4/\ln 3 = 2.26$. The most important

signature of hierarchical modularity is the scaling of the clustering coefficient, which follows C(k) ~ k –1 a straight line of slope –1 on a log–log plot. In literature, AI techniques such as neural networks have been used in construction of hierarchical network models (Rahmel, 1996).

3.2. Model Evaluation

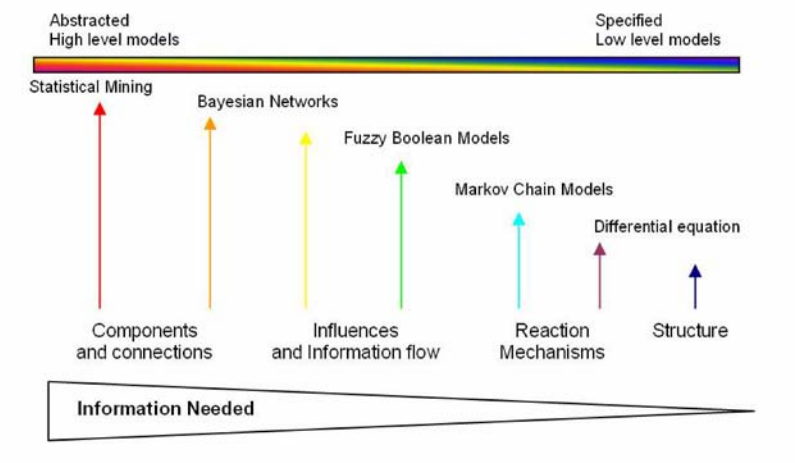
Computational model evaluation refers to finding out how closely the model constructed in phase 1 fits the data. Model evaluation methods can be very simple or highly complex. Figure 4 briefly shows various types of models with varying degree of abstraction of modelling and also the amount of information needed... The Fig. 4 is a result of conceptual comparison and hence is not empirical. Ordinary differential equations (ODEs) (de Hoon, 2003) models are used for small scale network that require least amount of information. Although gene regulations modelled via ODEs are successful to represent some reactions like linear production and degradation, they cannot describe the small system variability of the actual reactions. The system variability can be modelled using Markov chain models (Zhou et al, 2004) where there are sets of states and sets of state transitions along with their associated probabilities. Next level of abstraction for inferring genetic regulatory interaction is the well accepted boolean network model (Liang, 1998). Each gene is modelled as being either “ON” or “OFF” and the state of each gene at the next time step is determined by a boolean function of its inputs at the current time step. In a real cell, however, gene expression is a continuous variable. So, fuzzy models (Ram et al, 2006) have been proposed as alternative where membership functions for the expression levels are classified as, for example, high, medium and low. The fuzzy and boolean models are capable of modelling influences (activation/repression) and their corresponding direction of information flow, but require significant amount of data in order to deliver valid models.

Bayesian networks attempt to give a more accurate model of network behaviour, based on Bayesian probabilities for the variables (Heckerman, 1995). In the graphical representation of a Bayesian network, variables (genes) are represented as nodes and edges between nodes represent conditional dependence and causal relations. Bayesian networks include most of the previously proposed models as special cases and are inherently capable of incorporating existing knowledge. The Bayesian model is by far the most complicated and probably most efficient way of gene network modelling (Friedman, 2000) (Chickering et al, 1994) in the presence of noise. However, the accuracy of the results relies on quantity of data. Well established statistical data mining techniques can deliver the highest level of abstraction in modelling, however they are not tolerant to noise in the data. The domain expert’s acceptance of Bayesian network models is a good choice as it provides a graphical representation of a GRN and is facilitated by the stochastic and white-box nature of Bayesian networks (BNs). Furthermore, biologists find it easy to interpret. Recently, a comprehensive explanation and comparisons between the models’ evaluation strategies has been made available (Wessels et al, 2001).

3.3. Searching the Model Space

Most of the GRN modelling techniques involve searching for, from a large search space, the structure of the network given the data. This search problem is NP-hard (Chickering et al, 1994). For a dataset of n genes, there can *be* $n \times n$ arcs and the number of possible network structures amounts to $2n \times n$. For example, with n=10, the number of network reaches 1030. As n is typically of the order of thousands in a gene expression data, the number of possible structures explodes. Since exhaustive search in the

Figure 4. Abstraction of GRN modelling



structure space can be impractical and exact inference is NP-hard, approximation algorithms from computational intelligence domain are often necessary to obtain results. These stochastic optimisation techniques involve a large number of simulations with parametric variations resulting in a small subset of likely models. Some of the well known techniques for stochastic optimization are: 1) Genetic algorithm (GA) 2) Monte Carlo approach using metropolis sampling 3) GA, using bootstrap and 4) GA and Markov Chain Monte Carlo approach. Each of the resulting model represents the data with a reasonably good fit. The most difficult part of model fitting is discriminating between different model variants. If a particular model fits better then it is deemed, using appropriate statistical tests, that the model is the best model. The question is "How do we discriminate between several models, each of which fits the data reasonably well?" If the proposed model were in fact a genuine candidate, then we would expect the parameter space to coincide. This is based on the assumption that rate constants do not change as a result of external or internal perturbations.

4. AN ILLUSTRATIVE EXAMPLE MODEL

The search issues presented in previous section is further illustrated with the aid of a case study for GRN modelling. In this illustrative example, the network structure has a random topology, the network modeling is carried out by a causal modelling technique (Ram and Chetty, 2006) and the search is performed using a guided GA (GGA) (Ram and Chetty, 2006). It is suggested that the interested reader should refer to our earlier related publications for complete details. However in brief, the modelling steps are as follows:

4.1. Network Construction

The creation of GRN represented by the 'random network' is facilitated by an $n \times n$ connectivity matrix M. Each element in M, m_{ij} where i, j $\in$ {1, 2,…, n}, represents the edge between two genes, such that

$$m_{ij} = \begin{cases} 1, \text{ i} \rightarrow \text{j regulation is positive} \\ 0, \text{ otherwise} \\ -1, \text{ i} \rightarrow \text{j regulation is negative} \end{cases}$$

It can be noted that values of m_{ij} for $i=j$ can be ignored during the search process because it presents an edge between a node and itself. The network constructed is a static network and complexities of dynamicity, oscillations, and feedback loops are omitted for the sake of simplicity of the model evaluation process. A Poisson distribution is used and the constraint is imposed on the number of transcription factors per gene to be between 1 and 15.

4.2. Simulation

The candidate network model is automatically converted into the set of regulatory interactions required to perform the simulation of its static behaviour (it can be easily extended to include dynamics). This mathematical model can then be simulated by using continuous deterministic methods. Techniques for carrying out hybrid stochastic-deterministic simulations need investigations.

4.3. Network Validation

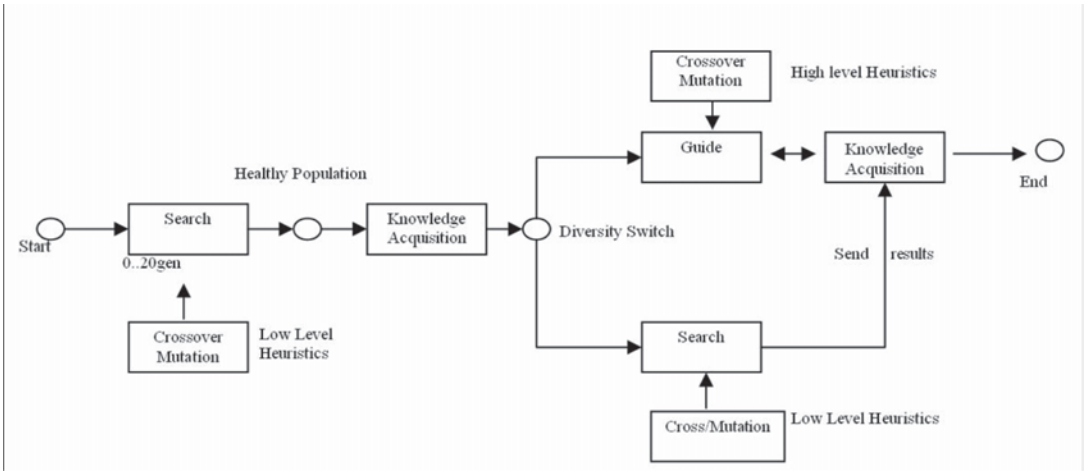
Given a candidate gene network, the fitness of the data is expressed as a set of scoring functions (Ram and Chetty, 2007). This includes fitness of the structure, direction of causality and sign (positive/negative) of regulation. While evaluating the fitness, the putative network is actually decomposed into Markov Blankets (MB) (Ram and Chetty, 2007) (comprising of the parents, the children, and the parents of the children of the node of interest) and conditional independence tests are applied in order to detect whether or not connections are direct or indirect. A direct regulation takes place where a TF regulates a target gene through its binding site. e.g., TF1 → regulated gene while the indirect regulation takes place when a TF regulates a target gene indirectly by regulating another TF, which regulates the target gene. For example, TF1 → TF2→ regulated gene. The direction and sign of regulation are recovered by estimating the time delay and time correlation between expression profiles of pairs of genes. The summation of the fitness of MBs is the fitness of the network. The candidate network which scores the highest total fitness best fits the data. The scoring functions can further be optimized using neural networks or fuzzy logic.

4.4. Evolution (Searching the Space of Models)

A guided genetic algorithm (GGA) (Ram and Chetty, 2007) is applied to create and evolve different networks to eventually obtain a network that best fits the microarray data.

The model selection is essentially a search procedure for finding the most plausible model fitting the observations. Given a network, the search algorithm qualitatively scores the structures derived by applying the local refinement operators (such as adding and deleting arcs), based on a proposed measure. The search keeps only the best structural modification (refinements) for the next iteration. The greedy approaches are not suitable for search as they do not guarantee global optima. Further, in BNs, a look-ahead approach (a greedy search is a 0-step look-ahead) is equivalent to multiple changes at once to the parent set of a node. Moreover, look-ahead approach has a very high computational cost that is

Figure 5. Novel guided strategy



intractable for our problem size. In our prior work (Ram and Chetty, 2007), a guided genetic algorithm (GGA) technique provided the necessary benefits of look-ahead with multiple changes to a node's Markov blanket graph (MBG) with a reduced computational cost.

This technique is applied in the present work as it relies on the observation that relationships which are conditionally independent (CI) under a specific MBG may not be CI under another MBG enabling decisions to be taken during the search to maximize the activity of each node. The working of the algorithm is illustrated in Fig. 5. When used with noisy synthetic datasets, it is experimentally observed that the technique is able to accurately learn functions that were best CI under uniform distribution even if data had only limited samples. Further, since the computational cost for the GGA has a linear relationship with the number of variables, it is better compared with the standard greedy learning algorithms which has computational cost complexity of the order $n \times n$. In the present work, we extend the GGA structure learning algorithm by including the frequent MBG finding technique where the aim is to find local maxima for MB of each node in the network. For a given GA simulation, all MBs of a given node from different networks form a set of MBGs called as graph transactions. Such a set of graph transactions for a node can typically be of the order of 200,000. The frequency of occurrence of a particular subgraph is determined by the number of graph transactions in which a subgraph occurs. Finding frequently occurring subgraphs over the entire set results in more quicker and better optimization of the GA and thus plays a critical role in the search process.

The steps involved in GGA are:

1. The initial phase (i.e. first 20 generations) of the main loop involves only low level heuristic operators, namely random crossover and mutation for generating population. To implement crossover, gene links from two members of the population are randomly selected and swapped between the pair of networks. Mutation is applied on individual networks by random deletion of an arc or random addition ensuring that a directed cycle is not created.
2. After the initial phase, a diversity switch selects between a low level and a high level GA heuristic operation. Diversity is a measure of variation between two individuals in a population. To calculate diversity, a mean skeleton network which is the basis for all networks for a population

is obtained. The difference in the number of edges calculated between an individual network in the GA population and the skeleton gives the diversity value. Briefly, an ongoing knowledge acquisition process keeps track of each node's Markov blanket that passes through the GA. These MBs are ranked according to their dominance (by means of individual fitness score) to appear in the final network. While adding or deleting an arc during a guided crossover and mutation, an arc score and path score are calculated from the individuals in the population. A Gaussian function is applied for producing partial randomness effect on the addition or deletion of an arc. The iteration repeats through the diversity switch and operators until the stopping criterion is reached.

3. Since the GA is stochastic in nature, the algorithm is repeated number of times (ten) and the resulting network structures are combined to reconstruct the final gene network.

The search is guided by exploiting certain characteristics of diversity and high level heuristics in order to generate good networks as quickly as possible. There are applications for ant colony optimization, swarm optimization, simulated annealing and other CI techniques that can also be investigated for their suitability in the search process.

4.5. Analysis and Post Processing

Methods for reducing the complexity of networks, and ensuring minimal connectivity are performed in this framework (Ram and Chetty, 2007). Issues like spurious arcs and time delay were not dealt with previously due to construction of random networks. We formulated a framework for path analysis as a post processing step after learning gene regulatory network where graph-theoretical measure of d-separation is applied on the final network and outliers are removed from the network. Similarly, delay paths are checked for their consistency. Computational intelligence techniques can be further investigated to speed up this process.

4.5.1 Synthetic Dataset

To establish the proposed methodology for successfully retrieving the structure of the underlying network, we evaluate its performance with the help of artificial and real-life yeast cell cycle data set. Using artificial data set generated by a novel synthetic data generator (Ram and Chetty, 2008), we investigate the accuracy and sample size requirements. In investigations involving real-life data set, we show that our approach is effective in inferring biologically significant interactions.

The expression data for the 40 gene synthetic network is generated by assuming that 6 genes have 3 regulatory inputs, 10 genes have 2 regulatory inputs, while the remaining genes have a single regulatory input. 33 interactions are designed to have a time delay of zero, 21 interactions have a time delay of one and 9 interactions have a time delay of two time points. Given this topology of the regulatory network, gene expression values are computed for each one of the 40 genes at 10 time points. The assumed network constituted 63 interactions with known regulatory weights and time delays associated with these interactions.

The algorithm was implemented in MATLAB and tested for the 40 gene artificial problem. To evaluate the proposed algorithm, we also executed the original genetic algorithm which incorporates low level heuristic operators alone. Table-1 shows the parameter setting of the GA. Figure 6 shows the plot of the best fitness value for the whole evolution process of 100 generations for guided GA and ordinary GA.

Table 1. GA parameter setting

Parameter	Value
Crossover probability	0.4
Elite Rate	0.1
Mutation probability	0.8 (evenly distributed over 4 types of mutation)
Population Size	150 – 200
Iterations	100
Diversity	0.7 (70%) (not applicable for ordinary GA)

From Fig.6, it can be seen that GA with guided strategy performs better than ordinary random GA after the initial 0- 20 generations. For the first 20 generations, the genetic algorithms in both the cases work in the same manner because adaptation can only be performed in later stages of the evolution. It can be noted that between generations 40 and 50, although the best value of fitness of the guided GA was lower than the ordinary GA, it later increased compared to random GA. This increase is attributed to the diversity switch. Fig. 7 shows the diversity measures throughout the evolution from generations 20 to 100, and a diversity measure of 0.98 was recorded during generations 40 and 50. As seen in nature, many genes of an organism stay inactive through its lifetime and are passed to further generations for later mutations or crossovers to activate, which is very well performed by the diversity switch and so this is seen as safeguarding diversity of the population. Both algorithms were restricted to runs of 100 generations. The guided GA converged at the top score at the 81st generation where all arcs were recovered while the simple GA stopped prematurely at the 55th generation.

From Fig.7, it can be noted that diversity switch has alternated from low level heuristics to high level heuristics a total of 9 instances falling above 0.7 (threshold) during the entire evolution of 100 generations.

Figure 6. Plot of best fitness values

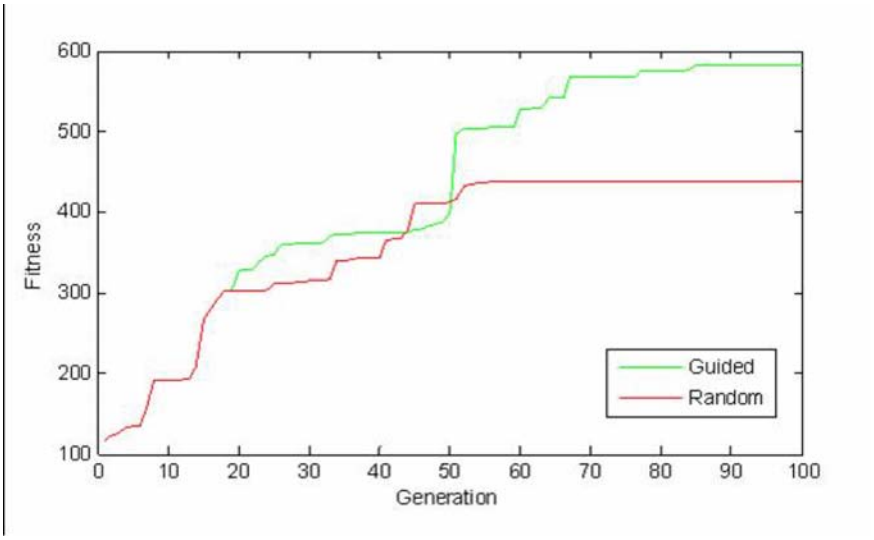


Figure 7. Diversity measures from generations 20 to 100

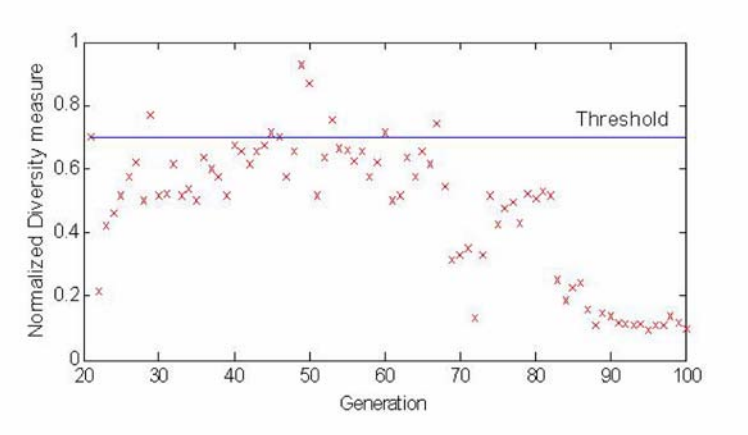


Fig. 8 shows the histogram of the normalized fitness values for the individual Markov blanket structures at the end of 100 generations. Normalization is done within the range from 0 to 1 based on the maximum and minimum value of fitness calculated during the evolution. The fitness in the range of 0.8 to 1 are alone taken into consideration as they constitute the best scores and it can be seen that most of the Markov Blankets inferred lie in the range of 0.99 and 0.96. This shows most of the sub-networks have achieved their maximum fitness values

Our results show that guided GA discovers causal GRN structures with a greater accuracy than existing standard genetic algorithms. This accuracy improvement does not come with an increase of search space. In all our experiments, 150 to 200 individuals are used in each of the 100 generations. Thus, a total of approximately 15,000 to 20,000 networks are searched in all to learn the causal structure. Considering the exhaustive search space is of 2^{n^2} networks, only a small percentage of the entire search space is needed by our algorithm to learn the causal structure.

Figure 8. Histogram of fitness values in the final generation

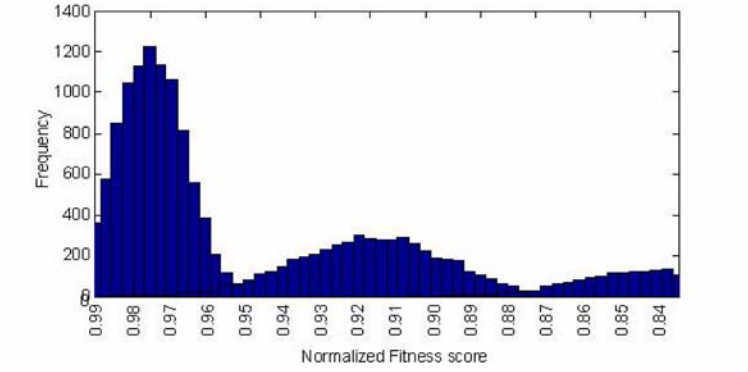
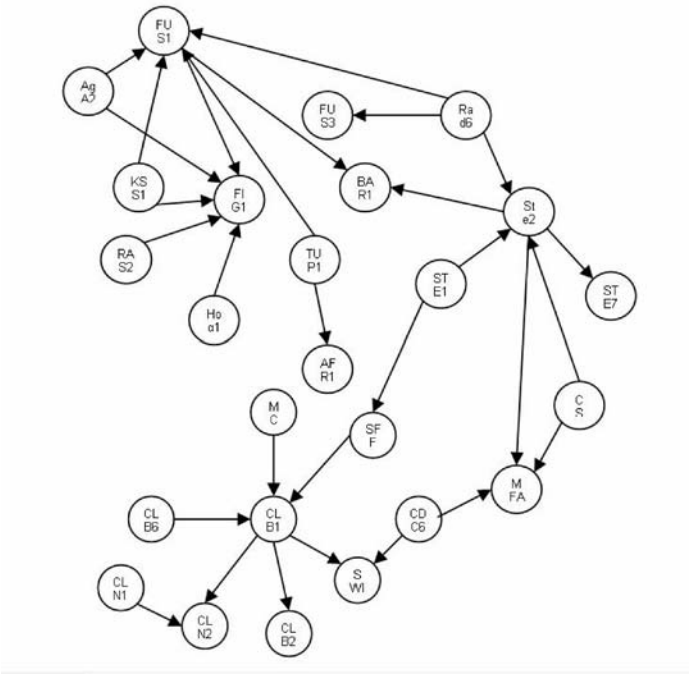


Figure 9. Reconstruction of GRN (subset of the 600 gene network)



4.5.2 Real Dataset

We further applied our approach to cell cycle expression data (Spellman et al., 1997), containing 76 gene arrays of 6177 S. cerevisiae ORFs. Gene expression levels are taken as continuous values. We are making use of all 76 samples from cdc15, alpha-factor and cdc28 datasets to determine the gene network structure. Since regulatory and signal relationships are currently not sufficiently known and because of the size limitation imposed by the relatively small number of time points, the analysis was restricted to 600 genes involved in the major cell cycle pathways given (Spellman et al., 1997). This modelling methodology for inferring gene regulatory network was applied to a subset of the genes from time series Saccharomyces cerevisiae (yeast) microarray dataset.

In Fig. 9, the reconstruction of the arcs (based on our causal modelling approach) for the genes involve the yeast cell cycle pathways such as spindle formation and cell cycle start. In this study, we consider

Table 2. Analysis details

No. of genes selected for analysis	600 genes
Limit on the TF's per gene	1 to 15 Tf's per gene
No. of interaction recovered	6300 interaction with highest confidence > 0.89
~3 TF's per gene	150-200
(4-9 TF's per gene)	Remaining
No. of GA generations	100

the group of important genes which includes CLB 1-6, CLN1-3, FUS3, KSU1, SIC1, SWI4, SWI6, TUP1 in cell-cycle regulation of *S.cerevisiae*. The validation is performed using a mathematical model of the cell-cycle events (Chen *et al*., 2002). Highly accurate regulatory interactions are *found* for the genes CLN1, CLN2, CLB1, CLB2, CLB5, SWI5 and SWI4. Some of the regulatory models have poor confirmation. The reason for this might be that some genes have much stronger signals during the *cdc15* experiment than during the other two. The genes CLN1 and CLN2 transcribing the G1 cyclins and the genes CLB5 and CLB6 transcribing the B-cyclins Clb5 and Clb6 are expressed in the G1-phase. Note the activator connections amongst the genes CLN1, CLN2, CLB5 and CLB6. The 'time delay' learning revealed the activator influences CLN1→CLN2, CLB6→CLB5, CLN1→CLB6 and CLN3→CLB6 (CLN3 is also the G1-specific cyclin).

The genes CLB1 and CLB2 are G2-specific cyclins, the gene SWI5 is the transcription factor also known to be expressed in G2-phase. Note the activator connections between the genes CLB1, CLB2 and SWI5. The 'time delay' problem inferred the activator regulation CLB1→SWI5 and CLB1→CLB2 (the 'time delay' 'AND' model suggested SWI5→CLB1, SWI5→CLB2). The inhibitory influences were inferred between the G1- and G2-specific genes confirming that the expression of these genes is separated in phases. The 'time delay' learning also revealed the inhibitory connections: CLB1, CLB2 → CLB5, CLB2→ CLN2, CLB6→CLB1, CLB6→CLB2, CLN3→SWI5, CLN3→CLB1 and CLN3→CLB2. The gene regulatory interactions described above and in the Fig.5 find support in the literature (Toyn et al, 1997).

5. CONCLUSION

In this chapter, modelling of gene regulatory network and the associated computational intelligence methodology required in its modelling is presented. .Various network configurations, namely random, scale free and hierarchical network are discussed. As a case study, we have presented important and relevant aspects of our work of causal modelling of GRN using guided genetic algorithm. The experiments are done on both synthetic data set as well as the real world yeast cell cycle data to demonstrate the performance of this novel approach. The modelling is validated using known results and the investigations reveal a number of interesting features. In future, work we will be focused on the effect of non coding RNA on regulation.

REFERENCES

Albert, R., & Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*, 47–91. doi:10.1103/RevModPhys.74.47

Batagelj, V., & Brandes, U. (2005). Efficient generation of large random networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 71*, 036113. doi:10.1103/PhysRevE.71.036113

Bay, S., Chrisman, L., Pohorille, A., & Shrager, J. (2004). Temporal aggregation bias and inference of causal regulatory networks. *Journal of Computational Biology, 11*, 971–985. doi:10.1089/cmb.2004.11.971

Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene networks: How to put the function in genomics. *Trends in Biotechnology, 20*, 467–472. doi:10.1016/S0167-7799(02)02053-X

Cancho, R. F., & Sole, R. V. (2001). The small-world of human language. *Proceedings. Biological Sciences*, *268*(1482), 2261–2265. doi:10.1098/rspb.2001.1800

Chen, K. C., Calzone, L., Csikasz-Nagy, A., Cross, F. R., Novak, B., & Tyson, J. J. (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, *15*, 3841–3862. doi:10.1091/mbc.E03-11-0794

Chickering, D. M., Geiger, D., & Heckerman, D. (1994). Learning Bayesian networks is NP-hard. (Tech. Rep. MSR-TR-94-17). Microsoft Research.

de Hoon, S., Imoto, K., & Kobayashi, N. Ogasawara, & Miyano, S. (2003). Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. *Pacific Symposium on Computation Biology*, 8, 17-28.

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*(1), 67–103. doi:10.1089/10665270252833208

de la Fuente, A., Bing, N., Hoeschele, I., & Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics (Oxford, England)*, *20*, 3565–3574. doi:10.1093/bioinformatics/bth445

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*, 799–805. doi:10.1126/science.1094068

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*, 601–620. doi:10.1089/106652700750050961

Gilbert, E. N. (1959)... *Annals of Mathematical Statistics*, *30*, 1141. doi:10.1214/aoms/1177706098

Heckerman, D., Geiger, D., & Chickering, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197–243.

Kishino, H., & Waddell, P. J. (2000). Correspondence analysis of genes and tissue types and finding genetic links from microarray data. In *Genome Inform Ser Workshop Genome Inform.* (pp. 83–95).

Kitano, H. (2001). *Foundations of systems biology*. Cambridge, MA: MIT Press.

Liang, S., Fuhrman, S., & Somogyi, R. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architecture. *Pacific Symposium on Biocomputing*, 3, 18–29.

McAdams, H. H., & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *94*, 814–819. doi:10.1073/pnas.94.3.814

Murphy, K., & Mian, S. (1999). Modelling gene expression data using dynamic Bayesian networks. Berkeley, CA: University of California.

Rahmel, J. (1996). SplitNet: A dynamic hierarchical network model. *AAAI/IAAI*, 2, 1404.

Ram, R., & Chetty, M. (2007). Learning structure of gene regulatory networks. *6th IEEE International Conference on Computer and Information Science* (pp. 525-531).

Ram, R., & Chetty, M. (2007). A guided genetic algorithm for gene regulatory network. *Proc IEEE Congress on Evolutionary Computation* (pp. 3862-3869).

Ram, R., & Chetty, M. (2007). Framework for path analysis for learning gene regulatory network. *Pattern Recognition in Bioinformatics, Springer* (pp. 264-273).

Ram, R., & Chetty, M. (2008). Generating synthetic gene regulatory networks. *Pattern Recognition in Bioinformatics, Springer* (pp. 237-249).

Ram, R., Chetty, M., & Dix, T. I. (2006). Fuzzy model for gene regulatory networks. *Proc. IEEE Congress on Evolutionary Computation* (pp. 1450-1455).

Ram, R., Chetty, M., & Dix, T. I. (2006). Causal modelling of gene regulatory network. *Proc. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, (CIBCB)* (pp. 1-8).

Savageau, M. A. (1998). Rules for the evolution of gene circuitry. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *3*, 55–65.

Širava, M., Schäfer, T., Eiglsperger, M., Kaufmann, M., Kohlbacher, O., Bornberg-Bauer, E., & Lenhof, H. P. (2002). BioMiner-modeling, analyzing, and visualizing biochemical pathways and networks. *Bioinformatics (Oxford, England)*, *18*(Suppl. 2), S219–S230.

Spellman, P. T., & Sherlock, G. (1998). Comprehensive identification of cell cycleregulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, *9*(December), 3273–3297.

Thomas, R., & Thieffry, H. (1995). Dynamical behaviour of biological regulatory networks-I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, *57*(2), 247–276.

Toyn, J. H. (1997). The Swi5 transcription factor of Saccharomyces cerevisiae has a role in exit from mitosis through induction of the Cdk-inhibitor Sic1 in telophase. *Genetics*, *145*, 85–96.

Tyson, J. J., & Othmer, H. G. (1978). The dynamics of feedback control circuits in biochemical pathways. R. Rosen (Ed.) New York: Academic Press.

Van Someren, E. P., Wessels, L. F., et al. (2001). Genetic network models: A comparative study. *Proc. of SPIE, Micro-arrays: Optical Technologies and Informatics*.

van Someren, E. P., Wessels, L. F. A., Backer, E., & Reinders, M. J. T. (2002). Genetic network modelling. *Pharmacogenomics*, *3*(4), 1–19.

Wang, X. F., & Chen, G. R. (2003). Complex networks: Small-world, scale-free, and beyond. *IEEE Circuits and Systems Magazine*, *3*(1), 6–20. doi:10.1109/MCAS.2003.1228503

Wolf, D. M., & Eeckman, F. H. (1998). The relationship between genomic regulatory element organization and gene regulatory dynamics. *Journal of Theoretical Biology*, *195*, 167. doi:10.1006/jtbi.1998.0790

Zhou, X., Wang, X., Pal, R., Ivanov, I., Bittner, M. L., & Dougherty, E. R. (2004). A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics (Oxford, England)*, *20*(17), 2918–2927. doi:10.1093/bioinformatics/bth318

KEY TERMS AND DEFINITIONS

Bayesian Network (BN): Probabilistic network structure in which nodes represent variables (genes) and edges between nodes represent their interactions.

Causal Model: Bayesian model specifying direct and indirect cause and effect relationships amongst the variables.

Directed Acyclic Graph (DAG): Directed graph showing the structure of causal relationships without having any feedback loops.

Gene Regulatory Network (GRN): Network of gene-gene interactions with regulatory relationships.

Genetic Algorithm (GA): Evolutionary algorithm based on stochastic parallel search technique.

Markov Blanket (MB): For any node of a BN under consideration, MB consists of the parents, the children and the other parents of the children.

Synthetic datasetMicroarray: A technique for performing DNA experiments in parallel to measure mRNA abundance of genes in a genome of an organism.

NP Hard: A property of computational search problems in which these problems can be solved in polynomial time.

Section 4

Structure and Parameter Learning

Chapter 11
A Synthesis Method of Gene Regulatory Networks based on Gene Expression by Network Learning

Yoshihiro Mori
Kyoto Institute of Technology, Japan

Yasuaki Kuroe
Kyoto Institute of Technology, Japan

ABSTRACT

Investigating gene regulatory networks is important to understand mechanisms of cellular functions. Recently, the synthesis of gene regulatory networks having desired functions has become of interest to many researchers because it is a complementary approach to understanding gene regulatory networks, and it could be the first step in controlling living cells. In this chapter, we discuss a synthesis problem in gene regulatory networks by network learning. The problem is to determine parameters of a gene regulatory network such that it possesses given gene expression pattern sequences as desired properties. We also discuss a controller synthesis method of gene regulatory networks. Some experiments illustrate the performance of this method.

INTRODUCTION

Investigating gene regulatory networks is important to understand mechanisms and functions of organisms and many researchers have been studied them from various view points. Recently there have been increasing research interests in synthesizing gene regulatory networks and several studies have been done. Those studies are motivated by two ways. One is that the synthesis of gene regulatory networks could be the first step in controlling and monitoring biochemical processes in living cells. The other is that it is a complementary approach to investigating and understanding mechanisms of real gene regulatory networks, that is to say, by synthesizing simple artificial networks and analyzing their behavior

DOI: 10.4018/978-1-60566-685-3.ch011

and functions, one can get some insights into functions of real gene regulatory networks. For example, Elowitz and Leibler (2000); Fung et al. (2005); Tuttle, Salis, Tomshine and Kaznessis (2005) synthesize artificial gene networks having oscillatory behaviors. Analyzing the synthesized networks could give some insights into investigating and understanding oscillatory behavior of organisms, e. g. circadian rhythm. Another example is the study on synthesizing artificial networks having a toggle switch like function (Atkinson, Savageau, Myers and Ninfa (2003); Deans, Cantor and Collins (2007); Gardner, Cantor and Collins (2000)).

Recently, Ichinose and Aihara (2002); Nakayama, Tanaka and Ushio (2006) discuss a synthesis problem in gene regulatory networks having desired properties. In those studies the desired properties are given by expression pattern sequences which describe changes of expression levels of genes. Furthermore, Nakayama et al. (2006) discuss a controller synthesis problem, in which controller gene regulatory networks are synthesized so that an objective gene regulatory network has desired expression pattern sequences.

In this chapter, we discuss the same synthesis problem and the controller synthesis problem of gene regulatory networks, in which the desired properties are given by expression pattern sequences. We present a novel synthesis method by network learning (Mori, Kuroe and Mori 2006). Gene regulatory network models are generally described by nonlinear differential equations and it is difficult to derive a synthesis method directly from nonlinear differential equation models. In order to overcome this difficulty we derive discrete-time networks possessing the equivalent time evolutions of expression pattern sequences to those of the differential equation models. We formulate the synthesis problem as a learning problem of the discrete-time networks and an efficient algorithm to solve the learning problem is derived. If the differential equation models are given by the piecewise linear network model (Glass 1975) with some class of interaction functions, the derived discrete-time networks are equivalent to a class of recurrent high-order neural network(RHONN)s and the synthesis problem is reduced to a learning problem of RHONNs.

The presented synthesis method can be applied to more general models of gene regulatory networks than the model used in Ichinose and Aihara (2002); Nakayama, at el, (2006) and it can be also applied to various synthesis problems. For example, the synthesis method can be extended and applied to the synthesis of gene regulatory networks possessing multiple desired expression pattern sequences, cyclic expression pattern sequences and stable cyclic expression pattern sequences.

BACKGROUND

There are several other studies on analysis and synthesis of gene regulatory networks. For examples, Hasty and Isaacs (2001) consider the gene regulatory network models described by nonlinear differential equations based on chemical reactions and investigate parameter regions such that they possess oscillatory behavior. Rodrigo et al. (2007) propose a synthesis method of gene regulatory network models such that they possess desired behavior, e. g. logical functions. Guido et al., 2006, Weiss et al., 2003 discuss a method for synthesizing rather complex gene regulatory networks by using simple gene regulatory networks as parts of them.

It is expected that the presented synthesis method makes some contributions toward understanding and synthesizing gene regulatory networks. In gene regulatory networks, several models, from simpler ones to detailed ones, have been proposed (Jong 2002). A simplest model is the Boolean network model

or the Bayesian network model. Nonlinear differential equation model based on chemical reaction is a detailed model. It is much more difficult to analyze and synthesize gene regulatory networks by using such nonlinear differential equation models. In this chapter we use a differential equation model whose complexity is in the middle of that of the Boolean network model and the nonlinear differential equation models. It is, therefore, relatively easy to analyze it and derive a synthesis method. There have been done many theoretical studies for the Boolean network model because of its simplicity (e.g. Akutsu, Kuhara, Maruyama and Miyano 2003). It is expected that those theoretical results give a good insight into analysis and synthesis of the differential equation model.

SYNTHESIS PROBLEM

Problem Statement

There are several models of gene regulatory networks (Jong, 2002). In this chapter, we consider a continuous-time network model of gene regulatory networks, which is given by the following differential equations:

$$\dot{x}_i(t) = -g_i(x_i(t)) + f_i(w_{i1}, w_{i2}, \&, w_{im_i}, y_1(t), y_2(t), \&, y_n(t)), \quad x_i(0) = x_{i0} \tag{1}$$

$$y_i(t) = H(x_i(t)), \quad i = 1,2,\&,n, \tag{2}$$

where H is a threshold function:

$$H(x_i) = \begin{cases} 1 & \text{if } x_i \geq 0, \\ 0 & \text{if } x_i < 0, \end{cases} \tag{3}$$

n is the number of genes, $x_i(t)$ is a normalized expression quantity of the ith gene, $y_i(t) \in \{0,1\}$ is a binary variable describing the on/off information of expression of the ith gene, that is, $y_i(t)=1$ if the ith gene is expressed, $y_i(t)=0$ if the ith gene is not expressed, $f_i: \{0,1\}^n \rightarrow R$ is a nonlinear function describing an interaction among genes, w_{ij}, $j=1,2,\ldots,m_i$ are parameters of f_i, m_i is the number of them and $g_i: R \rightarrow R$ is a nonlinear function representing the degradation of the ith gene. In what follows, this model is represented in the vector form:

$$\dot{x}(t) = -g(x(t)) + f(w, y(t)), \quad x(0) = x_0 \tag{4}$$

$$y(t) = H(x(t)), \tag{5}$$

where $x=(x_1, x_2,\ldots, x_n)^T$, $y=(y_1, y_2,\ldots, y_n)^T$, $g=(g_1, g_2,\ldots, g_n)^T$, $f=(f_1, f_2,\ldots, f_n)^T$, $H(x)=(H(x_1), H(x_2),\ldots, H(x_n))^T$, $w=(w_1, w_2,\ldots, w_n)^T$ and $w_i=(w_{i1}, w_{i2},\ldots, w_{im_i})^T$. We suppose that any g_i has the inverse function g_i^{-1}. The interactions among genes depend on on/off information $y(t)$ of the expression of genes and therefore the interactions among genes change if one of the expression levels of $x_i(t)$, $i=1,2,\ldots,n$ crosses zero.

We call y an expression pattern and we say that a gene regulatory network (1), (2) has an expression pattern sequence:

$$y^{(0)} \rightarrow y^{(1)} \rightarrow \cdots \rightarrow y^{(p)}, \tag{6}$$

where p is the length of the sequence, if there exist t_r, r=0,1,…,p satisfying $0<t_0<t_1<\ldots<t_r<\ldots<t_p$ and an initial state x_0 of $x(t)$ such that the expression pattern $y(t)$ of the gene regulatory network (1), (2) changes at t_r and $y(t)= y^{(r)}$, r=0,1,…,p during $t_r \leq t<t_{r+1}$ for the trajectory $x(t)$ starting from x_0.

The synthesis problem of the gene regulatory networks (1), (2) discussed in this chapter is as follows.

Synthesis problem For the given expression pattern sequence:

$$y^{*(0)} \rightarrow y^{*(1)} \rightarrow \cdots \rightarrow y^{*(p)}, \tag{7}$$

determine parameters w of the interaction functions f such that the gene regulatory network (1), (2) has the desired expression pattern sequence (7).

Because it rarely happens that signs of multiple normalized expression quantities change at the same time in real gene regulatory networks, we assume that

$$\| y^{*(r+1)} - y^{*(r)} \|^2 = 1, \quad r = 0,1,\&, p-1, \tag{8}$$

where $\| x \| = \sqrt{\sum_{i=1}^{n} x_i^2}$ for $x \in \mathrm{R}^n$.

For an expression pattern $\hat{y}$, define a region $\Omega_{\hat{y}}$ in the space of $x(t)$ by:

$$\Omega_{\hat{y}} := \{x \in R^n \mid \hat{y} = H(x)\} \tag{9}$$

and a point $e(\hat{y})$ by:

$$e_i(\hat{y}) := g_i^{-1}(f_i(w_i, \hat{y})), \quad i = 1,2,\&, n \tag{10}$$

or in the vector form:

$$e(\hat{y}) := g^{-1}(f(w, \hat{y})), \tag{11}$$

where $g^{-1}:=(g_1^{-1},g_2^{-1},\ldots,g_n^{-1})^{\mathrm{T}}$. The expression pattern $y(t)$ is equal to $\hat{y}$ if the expression quantity $x(t)$ is in $\Omega_{\hat{y}}$. The point $e(\hat{y})$ is called a "virtual" equilibrium point. The reason that the term "virtual" is put comes from the following facts. The right hand side of the equation (1) becomes equal to zero if $y(t)= \hat{y}$ and $x(t)= e(\hat{y})$, but $y(t)$ becomes $\hat{y}$ only if $x(t) \in \Omega_{\hat{y}}$, that is, $\dot{x}(t)$=0 at $e(\hat{y})$ if $e(\hat{y}) \in \Omega_{\hat{y}}$ and, in general, $\dot{x}(t) \neq 0$ at $e(\hat{y})$ if $e(\hat{y}) \notin \Omega_{\hat{y}}$.

Now, we make the following assumption for the gene regulatory network (1), (2).

Assumption The following statement holds for any expression pattern $\hat{y}$. Let $x(t)$ be a trajectory starting from x_0 where $x_0 \in \Omega_{\hat{y}}$. If $e(\hat{y}) \in \Omega_{\overline{y}}$ where $\overline{y}$ satisfies $\| \hat{y} - \overline{y} \|=1$ and $\hat{y}_j \neq \overline{y}_j$ for some j, then there exists $t_1>0$ such that $x_j(t_1)=0$ and for $0 \leq t \leq t_1$, $x_i(t) \neq 0$, $\forall\, i \neq j$.

This assumption means that if the "virtual" equilibrium point $e(\hat{y})$ is in the region $\Omega_{\bar{y}}$ adjacent to the region $\Omega_{\hat{y}}$, then any trajectory $x(t)$ in the region $\Omega_{\hat{y}}$ evolves to the region $\Omega_{\bar{y}}$.

Note that there are several models satisfying the condition of this assumption. One of them is piecewise-linear network model (Glass, 1975). We show that this model satisfies the condition of the assumption in later section.

A simplest model is the Boolean network model or the Bayesian network model. Nonlinear differential equation model based on chemical reaction is a detailed model. The gene regulatory network model (1), (2) is described by nonlinear differential equations. The time evolution of the expression quantities are described by differential equations. The interactions among genes of the model (1), (2) can be described by logical functions because the interaction functions depend on the expression pattern $y(t)$, that is, on/off information of the gene expressions. This means that the complexity of the gene regulatory network model (1), (2) is in the middle of that of the Boolean network model and nonlinear differential equation model. Hence, it is relatively easy to analyze the gene regulatory network model (1), (2). Furthermore, we can get information in time domain and information of expression levels of genes, while Boolean network model lacks such information, that is to say, it is expected to get more detailed information by using this model, e. g. periods and amplitudes of oscillations. Hence, before wet experiments, predictive information about interactions in a gene regulatory network having desired properties could be obtained by using the synthesis method.

Synthesis Method

Problem Formulation as Optimization Problem

To solve the synthesis problem, we derive constraint conditions such that the gene regulatory networks (1), (2) possess the expression pattern sequence (7). Consider two expression patterns $\hat{y}$ and $\bar{y}$ which satisfy $e(\hat{y}) \in \Omega_{\bar{y}}$, $\| \hat{y} - \bar{y} \| = 1$ and $\hat{y}_j \neq \bar{y}_j$. If $x(t) \in \Omega_{\hat{y}}$ for some t, there exists $t_1 > 0$ such that $x_j(t_1) = 0$ due to Assumption for the gene regulatory network (1), (2) and $y(t)$ changes from $\hat{y}$ to $\bar{y}$ at t_1. The above discussion implies that the expression pattern $y(t)$ of gene regulatory network (1), (2) changes from $\hat{y}$ to $\bar{y}$ if the parameters w satisfy the constraint $e(\hat{y}) \in \Omega_{\bar{y}}$ (that is, $\bar{y} = H(e(\hat{y}))$). Hence, if the parameters w of a gene regulatory network satisfy constraints:

$$y^{*(r+1)} = H(e(y^{*(r)})), \quad r = 0, 1, \cdots, p-1, \tag{12}$$

then the gene regulatory network (1), (2) has the expression pattern sequence (7). Such parameters w are not unique. We formulate the synthesis problem as an optimization problem in parameters w, whose constraints are the equations (12):

$$\min_{w} J \quad \text{s.t.} \quad y^{*(r+1)} = H(e(y^{*(r)})), \quad r = 0, 1, \cdots, p-1, \tag{13}$$

where J is a cost function depending on w, which represents a measure of the complexity of the gene regulatory network (1), (2). In this chapter, we choose

$$J = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \left| w_{ij} \right|. \tag{14}$$

It is known that the choice of l^1 norm (14) could make an optimal solution w^* much more sparse (Ishikawa, 1996), that is, the number of nonzero elements of w becomes much larger than that of the Euclid norm $\|w\|_2 = \sqrt{\sum_i \sum_j w_{ij}^2}$. The number of nonzero elements of w corresponds to the number of interactions among the genes, therefore a simpler gene regulatory network with smaller number of interactions could be obtained by the choice of (14).

Learning Method for Synthesis

To solve the optimization problem (13), we introduce a discrete-time network described by:

$$x_i[k+1] = g_i^{-1}(f_i(w_i, y[k])), \tag{15}$$

$$y_i[k] = H(x_i[k]), \quad i = 1, 2, \&, n \tag{16}$$

or in the vector form:

$$x[k+1] = g^{-1}(f(w, y[k])), \tag{17}$$

$$y[k] = H(x[k]). \tag{18}$$

Note that if the output $y[k]$ of the discrete-time network (15), (16) are equal to $y^{*(k)}$ (i.e. $y[k]=y^{*(k)}$, $k=0,1,\ldots,p$), we can see that the conditions (12) are satisfied. This implies that the optimization problem (13) can be formulated as a network learning problem as follows.

Let $y[k,x_0]$ be the outputs of the discrete-time network (15), (16) starting from $x[0]=x_0$. Then we consider a network learning problem:

$$\min_w \hat{J} = J_1 + \beta J, \tag{19}$$

where

$$J_1 = \frac{1}{2} \sum_{k=1}^{p} || \, y[k, x_0] - y^{*(k)} \, ||^2, \tag{20}$$

where $x_0 \in \Omega_{y^{*(0)}}$ and β is a weighting coefficient. If $\hat{J} = 0$ for w^*, then $y[k,x0]$ becomes equal to $y^{*(k)}$. This implies that the following equations hold,

$$y_i^{*(k)} = H(x_i[k, x_0]) \tag{21}$$

$$x_i[k, x_0] = g_i^{-1}(f_i(w_i, y_i^{*(k)})), \quad k = 1, 2, \&, p, \tag{22}$$

where $x_i[k,x_0]$ is the state of the discrete-time network (15), (16) at time k starting from the initial state $x[0]=x_0$. Hence the constraint conditions (12) are satisfied if a solution w^* achieves $\hat{J}=0$ in the learning problem (19).

The learning problem (19) can be solved by the gradient based methods such as the steepest decent method, the conjugate gradient method, the quasi-Newton method and so on. To calculate the gradient of the function $\hat{J}$, we replace the threshold function H in the discrete-time network (15), (16) by a smooth function S which can closely approximate to H. The discrete-time network (15), (16) becomes

$$x[k+1] = g^{-1}(f(w, y[k])), \tag{23}$$

$$y[k] = S(x[k]), \tag{24}$$

where $S(x)=(S(x_1), S(x_2),\ldots, S(x_n))^{\mathrm{T}}$. A learning algorithm is given as follows:

1. Choose initial values $w^{(0)}$ of w, and initial states of x so that $x_0 \in \Omega_{y^{*(0)}}$. Solve the discrete-time network (23), (24) and obtain $y[k,x_0]$, $k=1,2,\ldots,p$. Calculate $\hat{J}^{(0)}$ by using them. Set $\alpha=0$.
2. Compute the gradient $\partial \hat{J} / \partial w_{ij}$. Increment α; $\alpha=\alpha+1$.
3. Update w: $w^{(\alpha)}$ by using a gradient based method. Solve the discrete-time network (23), (24) and obtain $y[k,x_0]$, $k=1,2,\ldots,p$. Update $\hat{J}$: $\hat{J}^{(\alpha)}$ by using them.
4. If $|\hat{J}^{(\alpha)} - \hat{J}^{(\alpha-1)}|$ is small enough, stop, else go to Step 2.

Note that algorithms to compute the gradient $\partial J/\partial w_{ij}$ can be obtained based on the sensitivity analysis method by using adjoint equations or sensitivity equations.

The above discussion is summarized as follows. In order for a gene regulatory network (1), (2) to possess the desired expression pattern sequence (7), it is necessary for a trajectory $x(t)$ to traverse the regions $\Omega_{y^{*(r)}}$, $r=0,1,\ldots,p$. Due to Assumption for the gene regulatory network model (1), (2), the trajectory of the model (1), (2) depends on the allocation of the "virtual" equilibrium points $e(y)$'s, that is, where the virtual equilibrium points $e(y)$'s are in the state space. Using this property of the model, the synthesis problem is formulated as a problem of finding parameters satisfying the conditions (12). Note that the parameters satisfying the conditions (12) are not unique and therefore this problem is formulated as the optimization problem (13). The optimization problem is reduced to the discrete-time network learning problem (19) by introducing a discrete-time network (15), (16) which describes the allocation of the "virtual" equilibrium points $e(y)$'s.

Application to Piecewise Linear Network Model

Problem Formulation as a Learning Problem of Recurrent High-Order Neural Networks

In this section, we apply the synthesis method to the piecewise linear network model of gene regulatory network (Glass, 1975) which is one of the representatives of gene regulatory network models:

$$\dot{x}_i(t) = -d_i x_i(t) + f_i(w_i, y(t)), \tag{25}$$

$$y_i(t) = H(x_i(t)), \quad i = 1,2,\&,n \tag{26}$$

where d_i>0 is related to the degradation rate of the ith gene. In what follows, this model is represented in the vector form:

$$\dot{x}(t) = -Ax(t) + f(w, y(t)), \tag{27}$$

$$y(t) = H(x(t)), \tag{28}$$

where A is a diagonal matrix: A=diag(d_1,d_2,…,d_n). The interaction function f_i is usually defined as:

$$f_i(y) = a^{(i)} + \sum_{r=1}^{n} a_r^{(i)} y_r + \sum_{r=1}^{n-1} \sum_{s=r+1}^{n} a_{rs}^{(i)} y_r y_s + \cdots + a_{12\cdots n}^{(i)} y_1 \cdots y_n. \tag{29}$$

This model is obtained if we choose $d_i x_i$'s as $g_i(x_i(t))$ in the equation (1). Therefore, the piecewise linear network model (25), (26) is a subclass of the model (1), (2). This model satisfies Assumption. Consider two expression patterns $\hat{y}$ and $\overline{y}$ satisfying the following conditions: $\| \hat{y} - \overline{y} \|^2$=1, $\hat{y}_j \neq \overline{y}_j$ for some j and $e(\hat{y}) \in \Omega_{\overline{y}}$. Let the initial state x_0 of $x(t)$ be $x_0 \in \Omega_{\hat{y}}$. The conditions imply that sign(x_{0i})=sign(e_i($\hat{y}$)) for $i \neq j$ and sign(x_{0j}) ≠sign(e_j($\hat{y}$)) where sign(•) is the sign function. In the region $\Omega_{\hat{y}}$, a trajectory from x_0 is described as:

$$x_i(t) = e_i(\hat{y})(1 - \exp(-d_i t)) + x_{0i} \exp(-d_i t). \tag{30}$$

Therefore, $x_i(t) \neq 0$, $i \neq j$ for any t>0 because of sign(x_{0i})=sign(e_i($\hat{y}$)) for $i \neq j$ and there exists t_1>0 such that $x_j(t_1)$=0 owing to sign(x_{0j}) ≠sign(e_j($\hat{y}$)). Thus, Assumption is satisfied.

Now, the synthesis problem of gene regulatory network (25), (26) with the interaction functions (29) can be reduced to a learning problem of a class of RHONNs as follows. The synthesis problem is formulated as:

$$\min_{w} J \quad \text{s.t.} \quad y^{*(r+1)} = H(e(y^{*(r)})), \quad r = 0,1,\cdots,p-1, \tag{31}$$

where $e(y^{*(r)})=A^{-1}f(w,y^{*(r)})$. The discrete-time network corresponding to (23), (24) is:

$$x_i[k+1] = d_i^{-1} f_i(w_i, y[k]), \tag{32}$$

$$y_i[k] = S(x_i[k]), \quad i = 1,2,\&,n \tag{33}$$

or in the vector form:

$$x[k+1] = A^{-1} f(w, y[k]), \tag{34}$$

$$y[k] = S(x[k]). \tag{35}$$

Let $W_h=(w_1^{\mathrm{T}},w_2^{\mathrm{T}},\ldots,w_n^{\mathrm{T}})^{\mathrm{T}}$, $w_i=(w_{i1},w_{i2},\ldots,w_{im})$, $w_{i1}=a^{(i)}/d_i$, $w_{i2}=a_1^{(i)}/d_i,\ldots$, $w_{im}=a_{12\ldots n}^{(i)}/d_i$ where $m=2^n$. Then, the discrete-time network (32), (33) with the interaction functions (29) becomes equivalent to a class of RHONNs:

$$x[k+1] = W_h z[k], \tag{36}$$

$$y[k] = S(x[k]), \tag{37}$$

$$z[k] = (z_1(y[k]), z_2(y[k]), \cdots, z_m(y[k]))^T, \tag{38}$$

$$z_1 = 1, z_2(y) = y_1, z_3(y) = y_2, \cdots, z_{n+1}(y) = y_n, \cdots, z_m(y) = y_1 y_2 \cdots y_n,$$

in which y, z and W_h are outputs of neurons, outputs of high-order elements of y and weights for inputs z, respectively. In the next section, an efficient learning method by using adjoint networks is introduced.

An Efficient Learning Method by Using Adjoint Networks

Kuroe, Ikeda and Mori (1997) proposed a method for calculating the gradient of J_1 and derived an efficient learning algorithm by introducing adjoint networks for RHONNs. Therefore the synthesis problem can be solved by using the learning algorithm. Equations (36), (37), (38) can be formulated as:

$$z = (z_1(y), \cdots, z_m(y))^T, \tag{39}$$

$$u_i[k] = \sum_{j=1}^{m} w_{ij} z_j(y), \tag{40}$$

$$x_i[k+1] = u_i[k], \tag{41}$$

$$y_i[k] = S(x_i[k]). \tag{42}$$

The adjoint network for this network is defined by

$$\hat{x}_i[\tau] = \left.\frac{\partial S(x_i)}{\partial x_i}\right|_{x_i=x_i[k]} \left(\sum_{j=1}^{m} \hat{s}_{ji}[\tau] + \hat{y}_i[\tau]\right), \tag{43}$$

$$\hat{u}_i[\tau+1] = \hat{x}_i[\tau], \tag{44}$$

$$\hat{z}_i[\tau] = \sum_{j=1}^{n} w_{ji} \hat{u}_j[\tau], \tag{45}$$

$$\hat{s}_{ij}[\tau] = \left.\frac{\partial z_i(y_j)}{\partial y_j}\right|_{y_j=y_j[k]} \hat{z}_i[\tau], \tag{46}$$

where $\hat{y}[\tau]$ is the external inputs and $\tau=p-k$. If the initial state $\hat{u}[0]$ of the adjoint network is given by $\hat{u}[0]=0$ and the external inputs are given by

$$\hat{y}[\tau] = y[\tau, x_0] - y^{*(\tau)} \tag{47}$$

the gradient of the function J_1 for the parameter w_{ij} can be calculated by

$$\frac{\partial J_1}{\partial w_{ij}} = \sum_{k=1}^{p} z_j[k, x_0]\hat{u}_i[\tau, 0], \tag{48}$$

where $z_j[k,x_0]$'s are the signals of RHONNs (36), (37), (38) at k with initial state $x[0]=x_0$ and $\hat{u}_i[\tau,0]$'s are the signals of the adjoint network (43), (44), (45), (46) at τ with initial state $\hat{u}[0]=0$. The differentiation of function J at $w_{ij}=0$ is discontinuous. Hence, we define the gradient of function J for w_{ij} as:

$$\frac{\partial J}{\partial w_{ij}} = \begin{cases} 1 & \text{if } w_{ij} > 0 \\ 0 & \text{if } w_{ij} = 0 \\ -1 & \text{if } w_{ij} < 0 \end{cases} \tag{49}$$

A learning algorithm is given as follows.

1. Choose initial values $w^{(0)}$ of w, and an initial state of x[k] so that $x_0 \in \Omega_{y^{*(0)}}$. Solve the discrete-time network (36), (37), (38) and obtain $y[k,x_0]$, $x[k,x_0]$ and $z[k,x_0]$, k=1,2,…, p. Then calculate $\hat{J}^{(0)}$ by using them. Set α=0.
2. Set the initial value $\hat{u}[0]$=0, and the external inputs as (47). Solve the adjoint network (43), (44), (45), (46). Calculate the gradient $\partial J_1/\partial w_{ij}$ by the equation (48). Increment α; $\alpha=\alpha+1$.
3. Update w: $w^{(\alpha)}$ by using a gradient based method. Solve the discrete-time network (36), (37), (38) and obtain $y[k,x_0]$, $x[k,x_0]$ and $z[k,x_0]$, k=1,2,…,p. Then update $\hat{J}$: $\hat{J}^{(\alpha)}$ by using them.
4. If $|\hat{J}^{(\alpha)} - \hat{J}^{(\alpha-1)}|$ is small enough, stop, else go to Step 2.

Realization of Various Expression Pattern Sequences

The synthesis method can realize various expression pattern sequences in the gene regulatory network (1), (2). For realization of various expression pattern sequences, we show an extension of the synthesis method to realization of multiple desired expression pattern sequences. Let desired expression pattern sequences be given as:

$$y^{*(0,l)} \to y^{*(1,l)} \to \cdots \to y^{*(p_l,l)}, \quad l = 1, 2, \&, q \tag{50}$$

where q is the number of sequences and p_l is the length of the lth sequence. Constraint conditions for gene regulatory networks possessing the desired expression pattern sequences (50) are given by

$$y^{*(r+1,l)} = H(e(y^{*(r,l)})), \quad r = 0,1,\cdots,p_l - 1, \quad l = 1,2,\&,q. \tag{51}$$

This synthesis problem can be formulated as a network learning problem:

$$\min_{w} \hat{J} = J_1 + \beta J, \tag{52}$$

where

$$J_1 = \frac{1}{2}\sum_{l=1}^{q}\sum_{k=1}^{p_l} || \, y[k, x_0^{(l)}] - y^{*(k,l)} \, ||^2, \tag{53}$$

and $x_0^{(l)}$'s are initial values of $x[k]$ of the discrete-time network (15), (16) so that $x_0^{(l)} \in \Omega_{y^{*(0,l)}}$, l=1,2,…,q. This network learning problem is an extension of the network learning problem (19) and can be solved by a gradient based method.

Now we consider various synthesis problems in gene regulatory networks. For example, a synthesized gene regulatory network has a cyclic expression pattern sequence if $y^{*(0)}$=$y^{*(p)}$ in the desired expression pattern sequence (7). A gene regulatory network having two or more cyclic expression pattern sequences can be also synthesized by using the synthesis method with specifying cyclic expression pattern sequences in (50). Moreover, the synthesis method can realize stable cyclic expression pattern sequences. In this chapter, a "stable" expression pattern sequence is defined analogously as a stable limit cycle of continuous-time networks. Let a cyclic expression pattern sequence (7) be given and $\overline{\Omega}$ be the complementary set of the region $\cup_{r=0}^{p}\Omega_{y^{*(r)}}$. The cyclic expression pattern sequence (7) is stable if for any $x_0 \in \overline{\Omega}$, there exists $\hat{t}$ >0 such that a trajectory $x(t)$ of the gene regulatory network starting from x_0 enters $\cup_{r=0}^{p}\Omega_{y^{*(r)}}$ at $\hat{t}$ and the changes of expression pattern $y(t)$ equate with the cyclic expression pattern sequence (7) after $\hat{t}$. A stable expression pattern sequence can be realized by specifying behavior of the gene regulatory network (1), (2) in $\overline{\Omega}$. Hence, in addition to the constraints (12), the restrictions for the behavior of the gene regulatory networks in $\overline{\Omega}$ must be considered. For example, to synthesize a gene regulatory network having a stable expression pattern sequence:

$$(1,0,1)^T \rightarrow (0,1,1)^T \rightarrow (0,0,1)^T \rightarrow (1,0,1)^T, \tag{54}$$

it is sufficient to apply the synthesis method so that the gene regulatory network has the expression pattern sequence (54) and additional two following expression pattern sequences:

$$(0,1,0)^T \rightarrow (1,1,0)^T \rightarrow (1,1,1)^T \rightarrow (0,1,1)^T, \tag{55}$$

$$(1,0,0)^T \rightarrow (0,0,0)^T \rightarrow (0,0,1)^T. \tag{56}$$

Thus, the synthesis of gene regulatory networks having stable cyclic expression pattern sequences can be done by using the synthesis method.

Controller Synthesis Problem

In this section, we discuss the following controller synthesis problem. Let a gene regulatory network (1), (2) be a controlled object. The controller synthesis problem is synthesizing a controller gene regulatory network so that the controlled objective gene regulatory network (1), (2) has the desired expression pattern sequence (7). Let the controller gene regulatory network be described by

$$\dot{x}_{ci}(t) = -g_{ci}(x_{ci}(t)) + f_{ci}(w_{ci1}, w_{ci2}, \&, w_{cim_i}, y_{c1}(t), y_{c2}(t), \&, y_{cn_c}(t)), \tag{57}$$

$$y_{ci}(t) = H(x_{ci}(t)), \quad i = 1, 2, \&, n_c \tag{58}$$

or in the vector form:

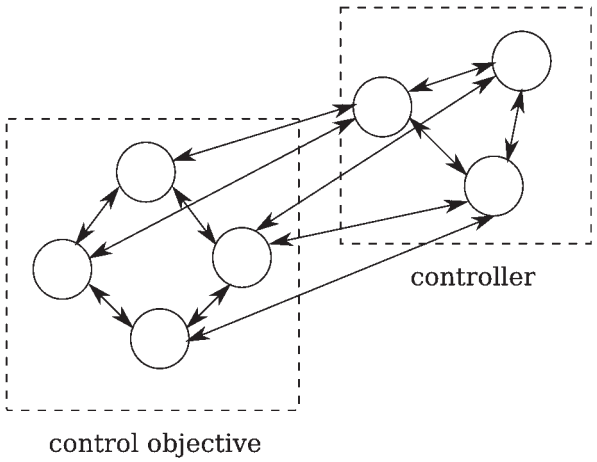
$$\dot{x}_c(t) = -g_c(x_c(t)) + f_c(w_c, y_c(t)), \tag{59}$$

$$y_c(t) = H(x_c(t)), \tag{60}$$

where the symbols with subscript *c* correspond to those without subscript *c* in (1), (2). We suppose that g_{ci} has the inverse function g_{ci}^{-1}. There are interactions among the genes of the objective gene regulatory network (1), (2) and those of the controller gene regulatory network (57), (58).

Fig. 1 is a schematic of the whole gene regulatory network. The whole gene regulatory network consisting of the objective gene regulatory network (1), (2) and the controller gene regulatory network (57), (58) is described by the following equations:

Figure 1. A gene regulatory network consisting of a controller and an objective gene regulatory network



$$\dot{x}_e(t) = \begin{pmatrix} \dot{x}(t) \\ \dot{x}_c(t) \end{pmatrix} = \begin{pmatrix} -g(x(t)) + f(w, y(t)) + \hat{f}(\hat{w}, \boldsymbol{y}(t), y_c(t)) \\ -g_c(x_c(t)) + f_c(w_c, y_c(t)) + \hat{f}_c(\hat{w}_c, y(t), y_c(t)) \end{pmatrix}, \tag{61}$$

$$y_e(t) = \begin{pmatrix} y(t) \\ y_c(t) \end{pmatrix} = \begin{pmatrix} H(x(t)) \\ H(x_c(t)) \end{pmatrix}, \tag{62}$$

where $\hat{f}$ and $\hat{f}_c$ are the interaction functions, $\hat{w}=(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_n)^{\mathrm{T}}$, $\hat{w}_i=(\hat{w}_{i1}, \hat{w}_{i2}, \ldots, \hat{w}_{i\hat{m}_i})^{\mathrm{T}}$, $\hat{w}_c=(\hat{w}_{c1}, \hat{w}_{c2}, \ldots, \hat{w}_{cn})^{\mathrm{T}}$, $\hat{w}_{ci}=(\hat{w}_{ci1}, \hat{w}_{ci2}, \ldots, \hat{w}_{ci\hat{m}_i})^{\mathrm{T}}$, $\hat{w}_{ij}$ j=1,2,…, $\hat{m}_i$ and $\hat{w}_{cij}$, j=1,2,…, $\hat{m}_{ci}$ are parameters of $\hat{f}_i$ and $\hat{f}_{ci}$, respectively.

Let the desired expression pattern sequence for the objective gene regulatory network (1), (2) be given as (7). The controller synthesis problem is:

Controller synthesis problem For the given expression pattern sequence (7) and the objective gene regulatory network (1), (2), determine the interactions $\hat{f}$, f_c and $\hat{f}_c$ of the whole gene regulatory network (61), (62), that is, determine $\hat{w}$, w_c and $\hat{w}_c$ such that the objective gene regulatory network (1), (2) in the whole gene regulatory network (61), (62) has the desired expression pattern sequence (7).

For an expression pattern $\hat{y}_e$, let define a region $\Omega_{\hat{y}_e}$ and $\hat{e}(\hat{y}_e)$ as follows:

$$\Omega_{\hat{y}_e} := \{x_e = (x^T, x_c^T)^T \in R^{n+n_c} \mid \hat{y} = H(x), \hat{y}_c = H(x_c)\}, \tag{63}$$

$$\hat{e}(\hat{y}_e) := \begin{pmatrix} e(\hat{y}, \hat{y}_c) \\ e_c(\hat{y}, \hat{y}_c) \end{pmatrix} = \begin{pmatrix} g^{-1}(f(w, \hat{y}) + \hat{f}(\hat{w}, \hat{y}, \hat{y}_c)) \\ g_c^{-1}(f_c(w_c, \hat{y}_c) + \hat{f}_c(\hat{w}_c, \hat{y}, \hat{y}_c)) \end{pmatrix}, \tag{64}$$

where $\hat{y}_e=(\hat{y}^{\mathrm{T}}, \hat{y}_c^{\mathrm{T}})^{\mathrm{T}}$. The region $\Omega_{\hat{y}_e}$ describes the region of state space in which the expression pattern $y_e(t)$ of the whole gene regulatory network (61), (62) is equal to $\hat{y}_e$.

Now, we make the assumption for the gene regulatory networks (61), (62).

Assumption The following statement holds for any expression pattern $\hat{y}_e$. Let $x_e(t)$ be any trajectory of the network (61), (62) starting from x_{e0} where $x_{e0} \in \Omega_{\hat{y}_e}$. If there exists $\overline{y}_e$ such that $\hat{e}(\hat{y}_e) \in \Omega_{\overline{y}_e}$, $\| \hat{y}_e - \overline{y}_e \| = 1$ and $\hat{y}_{ej} \neq \overline{y}_{ej}$ for some j, then there exists $t_1 > 0$ such that $x_{ej}(t_1) = 0$ and $x_{ei}(t) \neq 0$ for any t, $0 \leq t \leq t_1$, $x_i(t) \neq 0$, $\forall\, i \neq j$.

The synthesis method can be applied to the controller synthesis problem with slight modifications. Constraint conditions corresponding to (12) become

$$y^{*(r+1)} = H(e(y^{*(r)}, y_c^{(r)})), \tag{65}$$

$$y_c^{(r+1)} = H(e_c(y^{*(r)}, y_c^{(r)})), \quad r = 0, 1, \&, p-1. \tag{66}$$

Hence, the controller synthesis problem is formulated as an optimization problem to find parameters $\hat{w}$, w_c and $\hat{w}_c$ satisfying the constraints (65), (66):

$$\min_{\hat{w}, w_c, \hat{w}_c} J_c \quad \text{s.t.} \tag{67}$$

$$y_c^{(r+1)} = H(e_c(y^{*(r)}, y_c^{(r)})), \quad r = 0, 1, \cdots, p-1, \tag{68}$$

where J_c is a cost function depending on $\hat{w}$, w_c and $\hat{w}_c$. In this chapter we choose

$$J_c = \sum_{i=1}^{n} \sum_{j=1}^{\hat{m}_i} \left|\hat{w}_{ij}\right| + \sum_{i=1}^{n_c} \sum_{j=1}^{m_{ci}} \left|w_{cij}\right| + \sum_{i=1}^{n_c} \sum_{j=1}^{\hat{m}_{ci}} \left|\hat{w}_{cij}\right|. \tag{69}$$

Therefore, the controller synthesis problem can be formulated as a learning problem of discrete-time networks as follows. A discrete-time network:

$$x[k+1] = g^{-1}(f(w, y[k]) + \hat{f}(\hat{w}, y[k], y_c[k])), \tag{70}$$

$$x_c[k+1] = g_c^{-1}(f_c(w_c, y[k], y_c[k]) + \hat{f}_c(\hat{w}_c, y[k], y_c[k])), \tag{71}$$

$$y[k] = H(x[k]), \tag{72}$$

$$y_c[k] = H(x_c[k]) \tag{73}$$

is introduced to solve the optimization problem (67), (68). Let $y[k,x_0,x_{c0}]$ and $y_c[k,x_0,x_{c0}]$ be the outputs of the discrete-time network (70), (71), (72), (73) where x_0 and x_{c0} are initial states of x and x_c, respectively. Now, we consider a network learning problem:

$$\min_{\hat{w}, w_c, \hat{w}_c} \hat{J}_c = J_{1c} + \beta J_c, \tag{74}$$

where

$$J_{1c} = \frac{1}{2} \sum_{k=1}^{p} || \, y[k, x_0, x_{c0}] - y^{*(k)} \, ||^2, \tag{75}$$

$x_0 \in \Omega_{y^{*(0)}}$ and β is a weighting coefficient. This learning problem is the same as (19) except searching parameters and initial conditions. Searching parameters w are replaced with $\hat{w}$, w_c and $\hat{w}_c$. To solve the learning problem (74) by using gradient based methods, we obtain a discrete-time network by replacing the threshold function H with a smooth function S in the discrete-time network (70), (71), (72), (73):

$$x[k+1] = g^{-1}(f(w, y[k]) + \hat{f}(\hat{w}, y[k], y_c[k])), \tag{76}$$

$$x_c[k+1] = g_c^{-1}(f_c(w_c, y[k], y_c[k]) + \hat{f}_c(\hat{w}_c, y[k], y_c[k])), \tag{77}$$

$$y[k] = S(x[k]), \tag{78}$$

$$y_c[k] = S(x_c[k]) \tag{79}$$

and the learning algorithm is modified as:

1. Choose initial values $\hat{w}^{(0)}$, $w_c^{(0)}$ and $\hat{w}_c^{(0)}$ of $\hat{w}$, w_c and $\hat{w}_c$, respectively. Set initial values x_0 of $x[k]$ so that $x_0 \in \Omega_{y^{*(0)}}$ and initial values x_{c0} of $x_c[k]$ randomly. Solve the discrete-time network (76), (77), (78), (79) and obtain $y[k,x_0,x_{c0}]$ and $y_c[k,x_0,x_{c0}]$, k=1,2,…,p. Calculate $\hat{J}_c^{(0)}$ by using them. Set α=0.
2. Compute the gradient $\partial \hat{J}_c / \partial w_{cij}$. Increment α; α=α+1.
3. Update $\hat{w}$, w_c and $\hat{w}_c$: $\hat{w}^{(\alpha)}$, $w_c^{(\alpha)}$ and $\hat{w}_c^{(\alpha)}$ by using a gradient based method. Solve the discrete-time network (76), (77), (78), (79) and obtain $y[k,x_0,x_{c0}]$, k=1,2,…,p. Update $\hat{J}_c$: $\hat{J}_c^{(\alpha)}$ by using them.
4. If $|\hat{J}_c^{(\alpha)} - \hat{J}_c^{(\alpha-1)}|$ is small enough, stop, else go to Step 2.

We choose the initial states x_{c0} of $x_c[k]$ randomly because no expression pattern sequence is given for the controller gene regulatory network. But the expressions of genes of the objective gene regulatory network are affected by those of the controller gene regulatory network, whose expression levels are positive. Hence, we set at least one element of x_{c0} as positive. If learning results are not sufficient to satisfy the restrictions (65), (66), then we change the initial states x_{c0} and repeat the learning process.

NUMERICAL EXPERIMENT

We show numerical experiments to illustrate the performance of the synthesis method. We use the piecewise linear networks (25), (26) with the interaction functions (29) for numerical experiments. For a smooth function S, which approximates the threshold function H, we use a sigmoidal function:

$$S(x) = \frac{1}{1+\exp(-5x)} \tag{80}$$

in these numerical experiments. We assume that the parameters d_i's of genes are given as d_i =1 for i=1,2,…,n. The weighting coefficient β in the cost functions $\hat{J}$ or $\hat{J}_c$ is determined by trial and error. Initially, we set β as β=0.01/(nm/2) and modify it. The reason of choice of the value 0.01/(nm/2) is as follows. If only an element of $y[k]$ is different from a corresponding element of $y^{*(k)}$, that is, for some k_1 and j, $y[k_1] \neq y^{*(k_1)}$ where $y_j[k_1] \neq y_j^{*(k_1)}$, $y_i[k_1] = y_i^{*(k_1)}$, $i \neq$ j, and $y[k] = y^{*(k)}$, $k \neq$ k$_1$, then J_1=0.5. In addition, if half elements of w are equal to zero and the remaining elements of w are equal to 1, then J is equal to nm/2. If we use these values as a criterion, we can expect that J_1 is small enough when $\hat{J}$ becomes less than 0.5 with β=0.01/(nm/2).

Realization of a Cyclic Pattern Sequence

Let a desired cyclic expression pattern sequence be given as:

$$(1,1,0,0,1)^T \rightarrow (1,1,1,0,1)^T \rightarrow (1,0,1,0,1)^T \rightarrow (1,0,1,1,1)^T \rightarrow (1,0,1,1,0)^T$$
$$\rightarrow (0,0,1,1,0)^T \rightarrow (0,0,1,0,0)^T \rightarrow (0,0,0,0,0)^T \rightarrow (1,0,0,0,0)^T$$
$$\rightarrow (1,1,0,0,0)^T \rightarrow (1,1,0,0,1)^T. \tag{81}$$

A gene regulatory network consisting of 5 genes is synthesized because expression patterns have 5 elements. We set the weighting coefficient β in the cost function $\hat{J}$ as β=0.001 and initial states x_0 of $x[k]$ as x_0=(1.0,1.0,-1.0,-1.0,1.0)$^{\mathrm{T}}$ so that $x_0 \in \Omega_{(1,1,0,0,1)^{\mathrm{T}}}$. Applying the synthesis method, parameters of a gene regulatory network having the expression pattern sequence (81) are obtained. Figure 2 shows a plot of the cost function $\hat{J}$ as a function of learning step. The cost function $\hat{J}$ converges to zero in smaller number of learning steps.

It is confirmed that a gene regulatory network (25), (26), (29) using the obtained parameters has the desired cyclic pattern sequence (81). An example of simulation results of the synthesized gene regulatory network is shown in Figure 3,where initial state is the same as in the above learning process. The numbers placed at the bottom of Figure 3 represent the expression patterns of this gene regulatory network. Vertical dashed lines show boundaries where the expression pattern $y(t)$ of the gene regulatory network changes. It can be seen that the obtained gene regulatory network has the desired expression pattern sequence (81).

Realization of a Stable Cyclic Pattern Sequence

In this section, we show a realization experiment of a stable cyclic expression pattern sequence. Let a cyclic expression pattern sequence be given as:

$$(0,1,1,0,0)^T \rightarrow (0,1,1,1,0)^T \rightarrow (1,1,1,1,0)^T \rightarrow (1,1,1,1,1)^T \rightarrow (1,1,0,1,1)^T$$
$$\rightarrow (1,0,0,1,1)^T \rightarrow (0,0,0,1,1)^T \rightarrow (0,1,0,1,1)^T \rightarrow (0,1,0,0,1)^T$$
$$\rightarrow (0,1,1,0,1)^T \rightarrow (0,1,1,0,0)^T. \tag{82}$$

We consider eleven additional expression pattern sequences:

Figure 2. The cost function

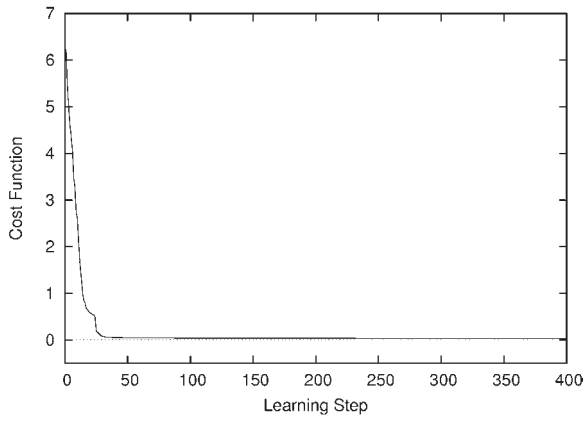
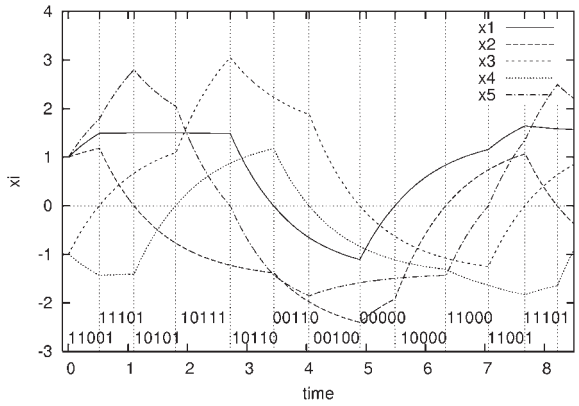


Figure 3. Simulation result of the obtained gene regulatory network: a cyclic pattern sequence



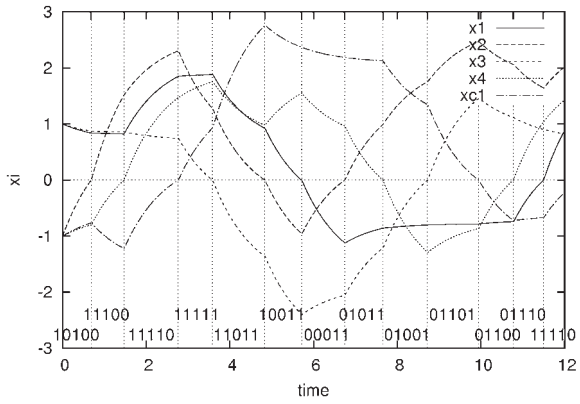
$$
\begin{aligned}
&(0,0,0,0,1)^T \rightarrow (0,1,0,0,1)^T,\\
&(0,0,0,1,0)^T \rightarrow (0,0,0,1,1)^T,\\
&(0,0,1,0,1)^T \rightarrow (0,1,1,0,1)^T,\\
&(0,0,1,1,1)^T \rightarrow (0,0,0,1,1)^T,\\
&(0,1,0,0,0)^T \rightarrow (0,1,0,0,1)^T,\\
&(0,1,0,1,0)^T \rightarrow (0,1,0,1,1)^T,\\
&(0,1,1,1,1)^T \rightarrow (1,1,1,1,1)^T,\\
&(1,0,0,0,1)^T \rightarrow (1,0,0,1,1)^T,\\
&(1,0,0,1,0)^T \rightarrow (1,0,0,1,1)^T,\\
&(1,0,1,0,0)^T \rightarrow (1,1,1,0,0)^T \rightarrow (1,1,1,1,0)^T,\\
&(1,1,0,1,0)^T \rightarrow (1,1,0,1,1)^T,
\end{aligned}
\tag{83}
$$

in order to stabilize the cyclic pattern sequence (82). A gene regulatory network consisting of 5 genes is synthesized. The number of expression pattern sequences is 12. The lengths of the expression pattern sequences are p_1=11, p_k=2,k=2,3,…,10, p_{11}=3, p_{12}=2. We set the weighting coefficient β in the cost function $\hat{J}$ as β=0.001 and 12 initial states $x_0^{(l)}$ of $x[k]$ so that $x_0^{(l)} \in \Omega_{y^*(0,l)}$, l=1,2,…,12. Applying the synthesis method, parameters of gene regulatory network having the expression pattern sequences (82), (83) are obtained. It is confirmed that the gene regulatory network (25), (26), (29) using the obtained parameters has the stable desired expression pattern sequence (82). We can see that for any $\hat{y}$, there exists some $\hat{t}$ such that the changes of expression pattern $y(t)$ of this gene regulatory network equate with the cyclic expression pattern sequence (82) after $\hat{t}$. An example of simulation results of the obtained gene regulatory network is shown in Figure 4,where initial state x_0 is chosen as x_0=(1.0, -1.0,1.0,-1.0,-1.0)$^{\mathrm{T}}$ so that $x_0 \in \overline{\Omega}$.

Controller Synthesis Experiments

We show three numerical experiments to evaluate the performance of the synthesis method. In these experiments, the objective gene regulatory networks are the same one and the desired expression patterns are the

Figure 4. Simulation result of the obtained gene regulatory network: a stable cyclic pattern sequence



same cyclic one. In experiment 1, a controller network consisting of one gene is synthesized. In experiment 2, the stability of the desired cyclic expression pattern sequence is considered as the desired property. In experiment 3, a controller network consisting of two genes is synthesized to stabilize the desired expression pattern sequence. We assume that the parameters d_{ci}'s of controller genes are given as d_{ci}=1 for i=1,2,…,n_c.

Controller Synthesis Experiment 1

In this numerical experiment, a controller gene regulatory network consisting of one gene is synthesized so that the objective gene regulatory network has the desired cyclic expression pattern sequence. The whole gene regulatory network consists of five genes. The desired expression pattern sequence is a cyclic sequence:

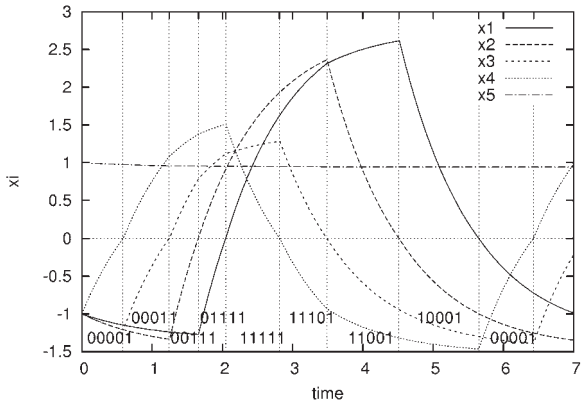
$$(0,0,0,0)^T \rightarrow (0,0,0,1)^T \rightarrow (0,0,1,1)^T \rightarrow (0,1,1,1)^T \rightarrow (1,1,1,1)^T$$

$$\rightarrow (1,1,1,0)^T \rightarrow (1,1,0,0)^T \rightarrow (1,0,0,0)^T \rightarrow (0,0,0,0)^T. \tag{84}$$

An objective gene regulatory network consists of four genes is given and the objective gene regulatory network doesn't have the desired expression pattern sequence (84). We set the weighting coefficient β in the cost function $\hat{J}_c$ as β=0.0001.

An example of simulation results of the whole gene regulatory network obtained by using the synthesis method is shown in Figure 5,in which the initial value of x_e is chosen as x_{e0}=(-1.0,-1.0,-1.0,-1.0,1.0^T. The binary numbers placed at the bottom of Figure 5 represent expression patterns of the whole gene regulatory network. It can be seen that the objective gene regulatory network has the desired expression pattern sequence (84).

Figure 5. Simulation result of the obtained gene regulatory network: controller synthesis experiment 1



Controller Synthesis Experiment 2

In this numerical experiment, the objective of control is stabilizing a cyclic expression pattern sequence of the objective gene regulatory network. The objective gene regulatory network in experiment 1 and the desired expression pattern sequence (84) are given. Let a controller gene regulatory network consist of one gene. The whole gene regulatory network consists of five genes. In this experiment, one of the desired properties is the stability of the cyclic expression pattern sequence (84). We synthesize a controller gene regulatory network so that the objective gene regulatory network has the expression pattern sequence (84) and

$$(0,1,0,0) \rightarrow (0,1,0,1)^T \rightarrow (1,1,0,1)^T \rightarrow (1,0,0,1)^T \rightarrow (1,0,1,1)^T$$

$$\rightarrow (1,0,1,0)^T \rightarrow (0,0,1,0)^T \rightarrow (0,1,1,0)^T \rightarrow (0,1,1,1)^T \tag{85}$$

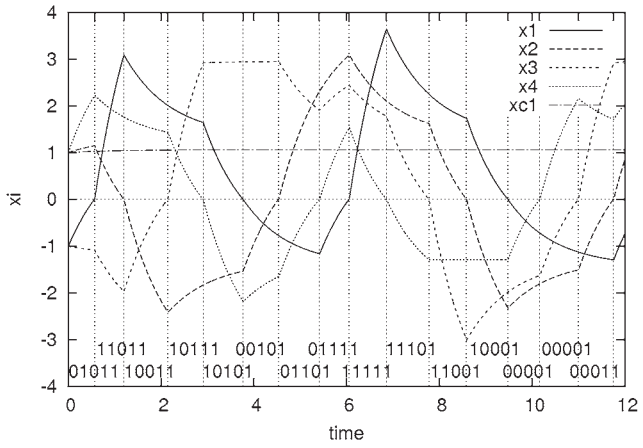
in order for the stability of the cyclic expression pattern sequence (84). Note that the two sequences (84) and (85) consist of all 16(=2^4) different patterns of the 4 bit binary vector $y(t)$ and the last pattern of the sequence (85) is the fourth pattern of the sequence (84). Hence, if the objective gene regulatory network has these sequences, the controller gene regulatory network can bring the stability of the cyclic expression pattern sequence (84). We set the weighting coefficient β in the cost function $\hat{J}_c$ as β=0.00001.

An example of simulation results of the whole gene regulatory network obtained by the controller synthesis method is shown in Figure 6,in which the initial value of x_e is chosen as x_{e0}=(-1.0,1.0,-1.0,1.0,1.0)^T. It can be seen that the objective gene regulatory network has the desired expression pattern sequences (84) and (85). Hence we can conclude that the objective gene regulatory network has the stable cyclic expression pattern sequence (84).

Controller Synthesis Experiment 3

In the above numerical experiments, controller gene regulatory networks consist of one gene. In this numerical experiment, we synthesize a controller gene regulatory network consisting of two genes. The

Figure 6. Simulation result of the obtained gene regulatory network: controller synthesis experiment 2



objective is to stabilize the desired cyclic expression pattern sequence (84). Let the same objective gene regulatory network consisting of four genes in experiment 1 be given. We choose two expression pattern sequences in order for the stability of the cyclic expression pattern sequence (84):

$$(0,1,0,0)^T \rightarrow (0,1,0,1)^T \rightarrow (1,1,0,1)^T \rightarrow (1,0,0,1)^T \rightarrow (1,0,1,1)^T \rightarrow (1,1,1,1)^T, \tag{86}$$

$$(1,0,1,0)^T \rightarrow (0,0,1,0)^T \rightarrow (0,1,1,0)^T \rightarrow (0,1,1,1)^T. \tag{87}$$

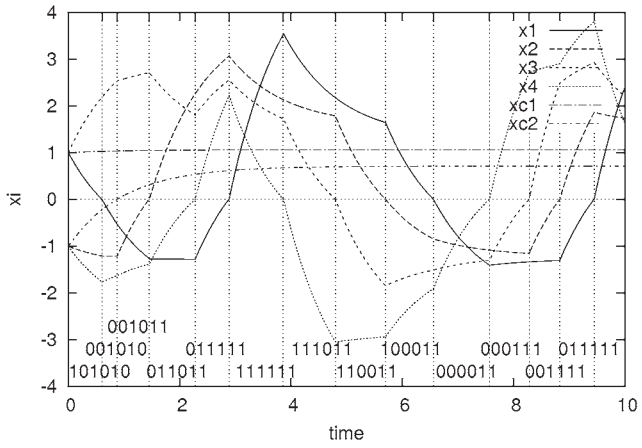
We set the weighting coefficient β in the cost function $\hat{J}_c$ as β=0.00001.

An example of simulation results of the whole gene regulatory network obtained by the synthesis method is shown in Figure 7,in which the initial value of x_e is chosen as x_{e0}=(1.0,-1.0,1.0,-1.0,1.0,-1.0)^T. It can be seen that the objective gene regulatory network has the desired expression pattern sequence (84) and the expression pattern sequence (87). We also observed that the objective gene regulatory network has the expression pattern sequence (86). Hence, we can conclude that the objective gene regulatory network has a stable cyclic expression pattern sequence (84).

FUTURE RESEARCH DIRECTIONS

The main objective of this chapter is to present a theoretically well-defined synthesis method of gene regulatory networks and to show how well the theory works by performing basic numerical experiments. The presented method could be applied to more realistic models of gene regulatory networks. In order to do that, it is necessary to explore the following problem. The presented method is applicable to the model (1), (2) of gene regulatory networks with any $g(x)$ and $f(x)$ satisfying the condition of Assumption. Therefore it is necessary to explore what kinds of functions are appropriate for $g(x)$ and $f(x)$ by using real data. This problem is a subject for the future work.

Figure 7. Simulation result of the obtained gene regulatory network: controller synthesis experiment 3



CONCLUSION

Synthesis of gene regulatory networks having desired functions is an important research area. In this chapter, we discussed the synthesis of gene regulatory network models possessing desired expression pattern sequences. We derived constraint conditions for parameters of a gene regulatory network so that the gene regulatory network possesses given expression pattern sequences. We showed that the synthesis problem can be formulated as a parameter optimization problem so that the constraints are satisfied with a solution of this optimization problem. We introduced a method for solving this parameter optimization problem by discrete-time network learning. For the piecewise linear network model with a class of interaction functions, the synthesis problem is reduced to a learning problem of a class of recurrent high-order neural networks.

It was shown that the synthesis method is successfully applied and can solve synthesis problems of various expression pattern sequences; e. g. a synthesized gene regulatory network possesses a cyclic expression pattern sequence, two or more cyclic expression pattern sequences or stable cyclic expression pattern sequences. We showed that the synthesis method of gene regulatory networks by network learning can be applied to the controller synthesis problem with slight modifications. We derived constraint conditions with respect to the parameters of the whole gene regulatory network consisting of an objective gene regulatory network and a controller gene regulatory network, and with respect to initial values of controller genes so that the controlled gene regulatory network possesses given desired expression pattern sequences. The controller synthesis problem was formulated as a parameter optimization problem.

Ichinose and Aihara (2002) proposed a synthesis method of gene regulatory network models (25), (26), (29) having desired expression pattern sequences. Nakayama et al. (2006) studied the controller synthesis problem and proposed a synthesis method of controller gene regulatory networks by introducing additional logical variables to the synthesis method proposed by Ichinose and Aihara (2002). Compared with these methods, the synthesis method in this chapter can be applied to both the synthesis problem and the controller synthesis problem with slight modifications. Furthermore the gene regulatory network model (25), (26), (29) is a subclass of the gene regulatory network model (1), (2) and the presented synthesis method can be applied to the synthesis problem of gene regulatory network model (25), (26), (29).

REFERENCES

Akutsu, T., Kuhara, S., Maruyama, O., & Miyano, S. (2003). Identification of genetic networks by strategic gene disruptions and gene overexpressions under a Boolean model. *Theoretical Computer Science, 298*, 235–251. doi:10.1016/S0304-3975(02)00425-5

Atkinson, M., Savageau, M., Myers, J., & Ninfa, A. (2003). Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. *Cell*, *113*, 597–607. doi:10.1016/S0092-8674(03)00346-5

Deans, T. L., Cantor, C. R., & Collins, J. J. (2007). A tunable genetic switch based on RNAi and repressor proteins for regulating gene expression in mammalian cells. *Cell*, *130*(2), 363–372. doi:10.1016/j.cell.2007.05.045

Elowitz, M. B., & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, *403*, 335–338. doi:10.1038/35002125

Fung, E., Wong, W. W., Suen, J. K., Bulter, T., Lee, S., & Liao, J. C. (2005). A synthetic gene-metabolic oscillator. *Nature*, *435*, 118–122. doi:10.1038/nature03508

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature*, *403*, 339–342. doi:10.1038/35002131

Glass, L. (1975). Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology*, *54*, 85–107. doi:10.1016/S0022-5193(75)80056-7

Guido, N. J., Wang, X., Adalsteinsson, D., McMillen, D., Hasty, J., & Cantor, C. R. (2006). A bottom-up approach to gene regulation. *Nature*, *439*(16), 856–860. doi:10.1038/nature04473

Hasty, J., & Isaacs, F. (2001). Designer gene networks: Towards fundamental cellular control. *Chaos (Woodbury, N.Y.)*, *11*(1), 207–220. doi:10.1063/1.1345702

Ichinose, N., & Aihara, K. (2002). A gene network model and its design. *The 15th Workshop on Circuit and Systems* (pp. 589-593) (in Japanese).

Ishikawa, M. (1996). Structural learning with forgetting. *Neural Networks, 3*, 509–521. doi:10.1016/0893-6080(96)83696-3

Jong, H. D. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*(1), 67–103. doi:10.1089/10665270252833208

Kuroe, Y., Ikeda, H., & Mori, T. (1997). Identification of nonlinear dynamical systems by recurrent high-order neural networks. In . *Proceedings of IEEE International Conference on Systems Man and Cybernetics*, *1*, 70–75.

Mori, Y., Kuroe, Y., & Mori, T. (2006). A synthesis method of gene networks based on gene expression by network learning. In *Proceedings of SICE-ICASE International Joint Conference* (pp. 4545–4550).

Nakayama, H., Tanaka, H., & Ushio, T. (2006). The formulation of the control of an expression pattern in a gene network by propositional calculus. *Journal of Theoretical Biology*, *240*, 443–450. doi:10.1016/j.jtbi.2005.10.014

Rodrigo, G., Carrera, J., & Jaramillo, A. (2007). Genetdes: Automatic design of transcriptional networks. *Bioinformatics (Oxford, England)*, *23*(14), 1857–1858. doi:10.1093/bioinformatics/btm237

Tuttle, L. M., Salis, H., Tomshine, J., & Kaznessis, Y. N. (2005). Model-driven designs of an oscillating gene network. *Biophysical Journal*, *89*, 3873–3883. doi:10.1529/biophysj.105.064204

Weiss, R., Basu, S., Hooshangi, S., Kalmbach, A., Karig, D., Mehreja, R., & Netravali, I. (2003). Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural Computing*, *2*(1), 47–84. doi:10.1023/A:1023307812034

KEY TERMS AND DEFINITIONS

Controller Synthesis Problem: Synthesis problem of controller gene regulatory networks.
Discrete-Time Network: Network described by difference equations.
Expression Pattern: Describe expression levels of genes.
Expression Pattern Sequence: Describe changes of expression levels of genes.
Gene Regulatory Network: Network describing interactions among genes.
Network Learning: Method solving optimization problem.
Synthesis Method: Method of synthesizing gene regulatory networks possessing desired behavior.

Chapter 12
Structural Learning of Genetic Regulatory Networks Based on Prior Biological Knowledge and Microarray Gene Expression Measurements

Yang Dai
University of Illinois at Chicago, USA

Eyad Almasri
University of Illinois at Chicago, USA

Peter Larsen
University of Illinois at Chicago, USA

Guanrao Chen
University of Illinois at Chicago, USA

ABSTRACT

The reconstruction of genetic regulatory networks from microarray gene expression measurements has been a challenging problem in bioinformatics. Various methods have been proposed for this problem including the Bayesian Network (BN) approach. In this chapter, we provide a comprehensive survey of the current development of using structure priors derived from high-throughput experimental results such as protein-protein interactions, transcription factor binding location data, evolutionary relationships, and literature database in learning regulatory networks.

DOI: 10.4018/978-1-60566-685-3.ch012

INTRODUCTION

The Bayesian Network (BN) has been proven to be a useful and important tool in biomedical applications such as clinical decision support systems (Beinlich, Suermondt, Chavez & Cooper, 1989), information retrieval (Baeza-Yates & Ribeiro, 1999), and discovery of gene regulatory networks (Friedman, Linial, Nachman & Pe'er, 2000). Automatic learning of BNs from observational data has been an area of intense research for more than a decade, yielding practical algorithms and tools (Spirtes, Glymour & Scheines, 1993). The ability of the BN approach to reconstruct genetic networks from microarray gene expression data has been extensively evaluated.

Consider a set of microarray experiments that measures the expression of a set of N genes over M different conditions. We denote the gene expression values by an $M \times N$ matrix $D = (d_1,\ldots,d_n)$. The BN method discovers a directed acyclic graph (DAG) S such that the posterior probability $P(S \mid X = D)$ is maximized. Here $X = (X_1,\ldots,X_N)$ denotes a set of random variables representing gene expression for genes $i = 1,\ldots,N$. Let π_i be the set of parents of node i in an acyclic network S. Then, the probability $P(X = D \mid S)$ can be decomposed into the product of local probabilities of nodes specified by the network structure S:

$$P(X = D \mid S) = \prod_{i=1}^{N} P(X_i = d_i \mid X_{\pi_i} = D_{\pi_i}), \tag{1}$$

where X_{π_i} denotes the subset of variables corresponding to π_i and D_{π_i} the corresponding observations. For ease of notation, we will omit the symbol X but use D indicating that X takes an observation D. The nodes in the learned network correspond to genes or their products and the edges correspond to direct probabilistic dependencies, such as causality, mediation, activation, or inhibition between the genes. The posterior probability $P(S \mid D)$ is proportional to the product of the likelihood $P(D \mid S)$ and the prior probability $P(S)$ of network structure S based on prior knowledge, i.e.,

$$\mathrm{P(S \mid D)} \propto \mathrm{P(S)P(D \mid S)} \tag{2}$$

The main approach to learning BNs from data is based on the strategy of search-and-score, which attempts to identify the most probable network S given the data D. This network has the highest posterior probability. Depending on assumptions, maximizing this probability corresponds to maximizing a score function. There are several ways to define the score. A straightforward definition is the likelihood $P(D \mid S)$. For discrete data and multinomial distribution, the K2 score (Cooper & Herskovits, 1992) is often used to evaluate the networks generated. For a given network S, this score is defined as the likelihood:

$$P(D \mid S) = \prod_{i=1}^{N} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!, \tag{3}$$

where N_{ijk} is the number of cases in D in which variable X_i has the kth value and the parent of i has the jth instantiation; q_i is the number of parents for i, and r_i the number of possible values of variable X_i. Thus,

$$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}\,!\,. \tag{4}$$

When the prior probability of S is considered, the K2 score for a network S can be modified by multiplying $P(S)$ to $P(D \mid S)$ in (3).

If a full Bayesian approach is preferred over the maximum likelihood, the score is the posterior probability in formula (2). In this case the computation of the marginal likelihood $P(D \mid S)$ requires a marginalization over the parameters θ:

$$P(D \mid S) = \int P(D \mid \theta_S, S) P(\theta_S \mid S) d\theta_S\,, \tag{5}$$

Where $P(\theta_S \mid S)$ is the prior distribution of the parameter θ_S over structure S. In a discrete BN structure S, the parameter θ_S defines a multinomial distribution for each variable X_i and each assignment of values to the parent of X_i. In a Gaussian BN structure S over continuous domains, then for each node, θ_S contains the coefficients for a linear combination of values of parent nodes and a variance parameter.

In general, the computation of the marginal posterior $P(D \mid S)$ is intractable in a full Bayesian approach. However, if certain regulatory conditions are satisfied and the data are complete, the integral over the parameter in (5) is analytically tractable. Two function families that satisfy these conditions are the multinomial distribution with a Dirichlet prior and the linear Gaussian distribution with a normal Wishart prior. For other general distributions, a Markov chain Monte Carlo (MCMC) scheme is often adopted to avoid the intractability of direct sampling from the posterior distribution (Friedman & Koller, 2003).

The general framework for search-and-score consists of two major steps:

Step1: search for a graph structure S;
Step2: evaluate the posterior probability $P(D \mid S)$.

For searching graph structures, different strategies can be considered. For example, the K2 algorithm (Cooper & Herskovits, 1992) requires the specification of an order of nodes from which the graph structures complying with the order are generated following a greedy strategy. Therefore, a procedure of producing candidate orders for the nodes is crucial to identify the optimal structure. Approaches to searching good orders include genetic algorithms (Larraaga, Poza, Yurramendi & Murga, 1996) and a MCMC method (Friedman & Koller, 2003). Since the order space is smaller than the structure space, it is more efficient to search the orders. On the other hand, a number of studies have demonstrated that the greedy search methods over a search space of DAGs works well (Heckerman, Meek & Cooper, 2006). The reader is referred to Chapter 'Bayesian Networks for Modelling and Inferring Gene Regulatory Networks' in this book for details on scoring metrics and search strategies. The major drawback of the search-and-score framework is the excessive computational cost, which can be partially alleviated by limiting the number of parents for each node (Friedman & Koller, 2003; Heckerman, Meek & Cooper, 2006). It may also get stuck in a local minimum.

Another category of BN learning algorithms are constraint-based methods (Chickering, 2002; Cooper, 1997; Singh & Valtorta, 1993). The constraint-based methods determine all dependence and independence relationships among variables through conditional independence test and construct networks that characterize these relationships. This is in contrast to the search-and-score methods which identify networks that fit data well. The constraint-based methods are computationally efficient, however, they

depend on a significance level for independency decision. They can be also unstable such that an early error in the search can result in different structures. Due to the high order conditional independence test, the constraint-based methods also require large sample size, which makes it unsuitable for the analysis of microarray data. A Hybrid learning method combining the constraint-based and the score-and-search methods has been proposed (Wang, Chen & Cloutier, 2007).

Methods of Using Prior Knowledge in Non-Biological Applications

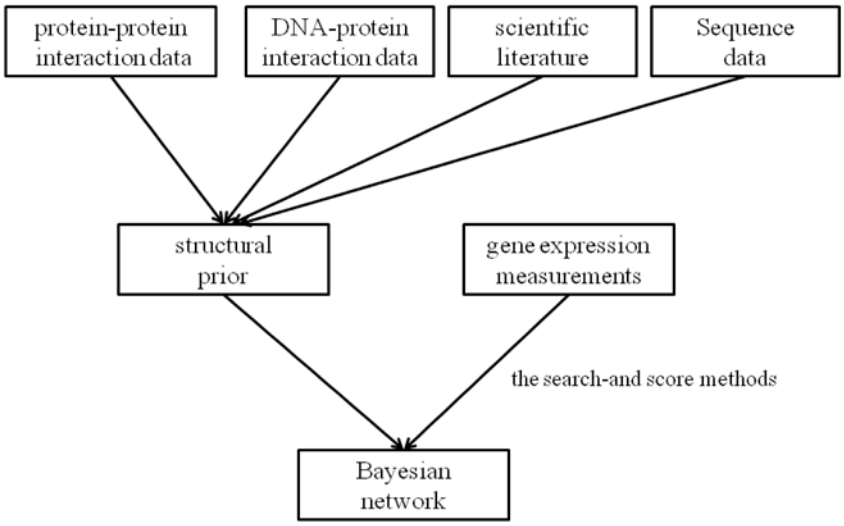
Since the number of graph structures is super-exponential to the number of variables (Neapolitan, 2003), multiple structures may achieve very similar high scores. It is, therefore, important to assemble prior knowledge to bias the search for a BN toward a model that contains the preference expressed in this prior. Prior knowledge in the form of a constraint or prior probability can reduce the search space, potentially improving the learning efficiency. In the BN research community, the prior over structures is usually considered the less important of the two components in BN learning: structure S and the parameters θ_S associated with the local probability distributions (Friedman & Koller, 2003). The simplest approach is to assume that each structure is equally likely and impose a uniform prior over all structures (Cooper & Herskovits, 1992; Heckerman, Meek & Cooper, 2006). The other alternative approach is to define a probability γ that each edge is present and subsequently assign networks with m edges a prior probability proportional to $\gamma^m(1-\gamma)^{\binom{N}{2}-m}$ (Wray, 1991).

Relatively little attention has been given to the use of additional expert knowledge that is not presented in the data. The expert knowledge ranges from logical constraints on the model structure (Campos & Castellano, 2007; Cheng, Bell & Liu, 1997; Cooper & Herskovits, 1992) or qualitative monotonicity relations between the variables (Heckerman, Geiger & Chickering, 1995) to prior distributions for network structures and parameterization of local dependencies (Campos & Castellano, 2007; Castelo & Siebes, 2000; Cooper & Herskovits, 1992; Iliopoulos, Enright & Ouzounis, 2001; Wray, 1991). Specially, Campos and Castellano (2007) investigated the effect of imposing restrictions on structures on BN learning. Three types of restrictions are considered: (1) existence of edges, (2) absence of edges, and (3) order restrictions. These types of restrictions are considered "hard" restrictions, as opposed to "soft' restrictions (Heckerman, Geiger & Chickering, 1995), in the sense that they are assumed to be true in all the candidate BNs.

Progress in Reconstruction of Transcription Regulatory Network Using Prior Knowledge

The early work in the reconstruction of gene networks used microarray data alone, largely ignoring existing prior biological knowledge (Akutsu, Miyano & Kuhara, 1999; Friedman, Linial, Nachman & Pe'er, 2000; Hartemink, Gifford, Jaakkola & Young, 2001; Imoto, Goto & Miyano, 2002; Imoto, Sunyong, Goto, & Aburatani, 2002; Pe'er, Regev, Elidan & Friedman, 2001). However, the difficulty with gene expression data is that complex interactions involving many genes have to be inferred from noisy data, which usually include small number of observations for each individual gene. In addition, since gene expression data only represent partial information from the transcription regulation mechanisms within a cell, the reconstructed networks often have poor accuracy (Husmeier, 2003). This suggests that the inclusion of complementary information in the BN learning is important (Van den Bulcke, Lemmens,

Figure 1. The framework of Bayesian network learning using prior knowledge



Van de Peer & Marchal, 2006). The general framework of using biological prior knowledge (Figure 1) consists of three steps:

(1) assemble biological knowledge into a structure prior probability or a set of constraints on networks;
(2) design a BN learning procedure on microarray data and the structure prior; and
(3) evaluate the confidence of the inferred interactions. Several representative approaches are summarized here.

Hartemink et al. (2002) used the information of transcription factor binding data gathered from a chromatin immuno-precipitation assay as constraints specifying which edges are required to be present and which are required to be absent in the network structure. This type of constraints on network structures is equivalent to a non-uniform prior over structures that give zero weight to models that include edges required to be absent. A slightly different version was also proposed to assign probability 1 to a directed edge from a regulatory to a target gene; probability 0 to the edge opposite to the above directed edge; and a constant probability to all the other edges (Gevaert, Van Vooren & De Moor, 2007).

Imoto et al. (2003) established a general framework that allows the systematic integration of gene expression data with other types of biological knowledge through a prior distribution over network structures. Using this framework, a consensus motif for a set of genes with a potential regulatory gene can be considered as prior knowledge. In a closely related work (Nariai, Kim, Imoto & Miyano, 2004), a list of protein-protein interactions was mined and used to construct a structure prior probability. Since this prior knowledge is of a very specific type, the biological implications of protein-protein interactions were exploited in the learning scheme by adding nodes representing protein complexes when the resulting structure better fits or explains the data. Similarly, evolutionary relationships between proteins (Tamada, Bannai, Imoto & Katayama, 2005) and pathway information from the Kyoto Encyclopedia

of Genes and Genomes (KEGG) database have been integrated using this framework (Imoto, Higuchi, Goto & Miyano, 2006). This framework has also been extended to allow the incorporation of multiple types of prior biological knowledge in network learning (Werhli & Husmeier, 2007).

High-throughput data, such as protein-protein binding interactions and transcription factor (Whitfield, George, Grant & Perou, 2006) binding locations in DNA sequence, inevitably include both false positives and false negatives. Usually, a *p*-value is provided determined for an identified interaction as a confidence measure. The integration of this type of knowledge requires considering the confidence of relationships. A probabilistic model that uses *p*-values to construct a structure prior was proposed (Bernard & Hartemink, 2005). In that model, the prior incorporates the *p*-values of transcription factor - DNA interactions given by ChIP experiments to weigh against evidence from expression data.

Scientific literature is also a source of prior knowledge. This type of knowledge can be collected by manual curation or automated procedures based on natural language processing algorithms and stored in databases. However, errors are inevitably introduced during the process due to either automated procedure or erroneous observations in the original publications. It would be desirable for the structure prior to reflect this aspect. A probabilistic framework of a joint learning model for repairing database errors and for estimating a gene network in the context of dynamic Bayesian network was provided (Imoto, Higuchi, Goto & Miyano, 2006). Another way to explore knowledge in literature is through text-mining methods (Gevaert, Van Vooren & De Moor, 2007). In that work, a gene can be represented by a normalized vector in which each entry shows the evidence in PubMed abstracts of this gene to a vocabulary based on the publicly available National Cancer Institute Thesaurus (http://nciterms.nci.nih.gov/NCIBrowser/). The cosine measure is used to obtain gene-to-gene distances, which is further used for the construction of structure priors.

Recently, Almasri et al. (2008) proposed another structure prior that incorporates novel knowledge mined from literature. In that work, prior knowledge of gene interactions was derived based on the statistical analysis of published interactions between gene/gene product pairs. The knowledge was represented by a likelihood score of interaction (LOI) for a pair of possible interacting genes and the corresponding *p*-value. This information was then explored (1) as a structure prior and (2) as constraints to reduce the search space in the BN algorithm.

PRIOR BIOLOGICAL KNOWLEDGE

We present a summary of high-throughput experiments based on which the structure priors were considered. In general, each method described here detects a set of interactions with only moderate overlap with the results of the other methods. This is not a failure of any single method, but indicates that the methods are complimentary and multiple approaches are required to identify the complete set of possible interactions in a system.

Experiments for Identifying Protein-Protein Interactions

Protein interactions can be analyzed by different genetic, biochemical, and physical methods. Some techniques screen a large number of proteins in a cell. The representatives are yeast two-hybrid (Y2H), tandem affinity purification (TAP), Mass Spectrometry (MS), DNA and protein microarrays, synthetic

lethality, and phage display. Other methods monitor and characterize specific biochemical and physico–chemical properties of a protein complex. The details on each method and databases for protein interactions can be found in a recent review paper (Shoemaker & Panchenko, 2007). Two methods, Y2H and MS, are selected here to give simple descriptions since the experimental data from these methods have been used in BN learning as prior knowledge.

The Y2H screen is used to identify pair-wise interactions between a 'bait' protein and a 'prey' protein (Chien, Bartel, Sternglanz & Fields, 1991). The Y2H method is highly sensitive, and does not require large amounts of fusion proteins to be expressed. Y2H can only detect pair-wise interactions between proteins. Interactions that require three or more proteins are not detected. In addition, interactions between membrane-localized proteins, which are unable to enter the nucleus and interact with DNA, cannot be detected. Protein interactions that require post-translational modification of one or both proteins cannot be detected either by this technology. The Y2H screen is prone to false positives. There are two major genome-wide data sets using yeast Y2H assays for the interrogation of genome-wide protein interactions (Ito, Ota, Kubota & Yamaguchi, 2002; Uetz, Giot, Cagney & Mansfield, 2000). Another technology, MS, can be used to identify proteins that bind together to form complexes (Gavin, Bosche, Krause & Grandi, 2002; Ho, Gruhler, Heilbut & Bader, 2002). MS requires proteins that have been broken into numerous smaller, more easily identifiable, polypeptides by proteolysis. These small protein fragments are identified in MS by their mass and charge. The identified protein fragments are reassembled into complete proteins by searching the identified pool of protein fragments against large-scale databases of proteins. MS can identify protein complexes rather than just binary protein-protein interactions. MS is, however, biased towards stable complexes and it requires that proteins are present at relatively high abundance (Ito, Ota, Kubota & Yamaguchi, 2002). Two protein interaction datasets resulted from the large scale MS studies are available (Gavin, Bosche, Krause & Grandi, 2002; Ho, Gruhler, Heilbut & Bader, 2002).

Chromatin Immunoprecipitated DNA on Microarray Chip (ChIP-chip)

Chromatin immunoprecipitation of DNA bound to proteins used for hybridization to oligonucleotide microarrays (ChIP-chip) can be used to identify DNA sequences that bind to specific proteins. Protein-DNA interactions such as transcription-regulator and DNA binding site or histone and chromatin can be found by this method (Lee, Rinaldi, Robert & Odom, 2002). Because ChIP-chip assays take place *en vivo*, this analysis method can detect DNA binding activity of a protein in its native state, including any condition-specific chromatin structure or post-translational modification necessary for the protein's function. The ChIP–chip assays allow the identification of the genomic region to which a particular protein is bound. However, because of limitations of the assays, it is difficult to identify the exact site within the region to which the protein is bound. Various computational methods have been proposed to allow flexible query and output options in databases that store experimental results. One can filter data sets to meet user-specified threshold on *p*-values for ChIP–chip data. The details on computational methods can be found (Elnitski, Jin, Farnham & Jones, 2006). The ChIP–chip data set (Lee, Rinaldi, Robert & Odom, 2002) has been used in various computational studies as the prior knowledge.

Knowledge from Literature

The automatic extraction of biological knowledge using natural language processing methods is becoming a useful tool for the survey of published work because of the sheer size of the body literature. However, the extracted information often includes errors. Furthermore, the extracted knowledge only represents the discoveries so far and does not provide new knowledge gained from the literature. To assess the possibility of interaction between a pair of genes/proteins, Larsen et al. (2007) utilizes the likelihood of interaction (LOI) scores derived from the systematic analysis of published interactions and their molecular functions based on Gene Ontology (GO) annotations (Ashburner, Ball, Blake & Botstein, 2000). The LOI-score is a measure of the likelihood that a gene or a gene product with a particular molecular function interacts with another gene or a gene product of a particular molecular function. More specifically, if two genes closely resemble by their molecular functions from previously observed interaction pairs, then they will be considered likely to interact. In their work, gene interactions for a set of yeast genes are first derived from an automated literature mining software. Then, each gene is annotated by the 23 GO Molecular Function (MF) annotations specified by the Saccharomyces Genome Database (SGD) GO Slim Mapper (Battle, Segal & Koller, 2005). Using a statistical procedure, each pair of GO MF annotations is assigned a LOI-score. Then the calculated LOI-scores for GO MF annotation pairs are used to generate LOI-scores for all possible gene interaction pairs in a set of query genes (Larsen, Almasri, Chen & Dai, 2007).

The framework for computing LOI-scores is general. LOI-scores need not to be limited to either the type of previously published interactions or GO MF for annotation of gene products. Any current large database of gene interactions could be used as the basis for LOI-score calculations and any appropriate gene product annotations could be used.

STRUCTURE PRIORS FROM BIOLOGICAL KNOWLEDGE

In this section, we introduce several frameworks for constructing structure prior probabilities from various types of prior biological knowledge.

Structure Prior Through the Framework of Energy Functions

As mentioned previously, a general framework for the construction of structure prior from biological knowledge has been proposed using the form of a Gibbs distribution (Imoto, Goto & Miyano, 2002; Imoto, Higuchi, Goto & Tashiro, 2003; Tamada, Kim, Bannai & Imoto, 2003). The prior biological knowledge is encoded through an energy function and an inverse temperature hyperparameter. More specifically, the network energy for a structure S is defined as follows:

$$E(S) = \sum_{(ij)\in S} E_{ij}, \tag{6}$$

where E_{ij} is the energy of an edge *(ij)* from gene *i* gene *j* in structure *S*. Within the BN framework, this total energy can be decomposed into the sum of the local energies:

$$E(S) = \sum_{j=1}^{N} \sum_{i \in \pi_j} E_{ij} = \sum_{j=1}^{N} E_j, \tag{7}$$

where π_j is the index set of parents of gene *j* in structure *S*, and $E_j = \sum_{i \in \pi_j} E_{ij}$ is a local energy defined by gene *j*. The probability of structure *S* is defined using the Gibbs distribution:

$$P(S \mid \beta) = \frac{1}{Z(\beta)} e^{-\beta E(S)}, \tag{8}$$

where $Z(\beta) = \sum_{S} e^{-\beta E(S)}$ is a partition function that takes sum over all structures *S* and β a positive hyperparameter associated with energy function. This parameter can be considered an indicator of the strength of the influence of the biological prior knowledge relative to the data. The prior distribution defined above becomes flat and uninformative about the network structure when $\beta \to 0$. Conversely, it becomes sharply peaked at the network structure with the lowest energy when $\beta \to \infty$. In this case, the highest prior probability is obtained.

In this framework the prior biological knowledge is summarized in the energy matrix $\{E_{ij}\}$. The entry E_{ij} takes value h_1 if there is an evidence for the presence of the edge from gene *i* to gene *j* from the biological knowledge or h_2 otherwise. Here, $0 < h_1 < h_2$. This type of definition is suitable for biological knowledge such as protein-DNA interactions in which gene *j* is regulated by gene *i*. However, for the protein-protein interaction, both E_{ij} and E_{ji} should take the same value h_1 if there is evidence of interaction. This definition can also represent prior knowledge from DNA or protein sequences. Genes that are regulated by a transcription regulator might have a consensus motif in their promoter DNA sequences. If genes $j_1,\ldots,j_n$ have a consensus motif for gene *i*, then one can set $E_{ij_k} = h_1$ and $E_{j_k i} = h_2$ for all $k = 1,\ldots, n$. The prior probability of structure *S* can be rewritten as:

$$P(S \mid \beta) = \frac{1}{Z(\beta)} \prod_{j=1}^{N} \prod_{i \in \pi_j} e^{-\beta E_{ij}}. \tag{9}$$

In this approach, the values of $\zeta_1 = \beta h_1$ and $\zeta_2 = \beta h_2$ are determined through the minimization of a scoring function. It was also noted that this general framework can take biological knowledge extracted from large body of literature (Imoto, Higuchi, Goto & Tashiro, 2003).

Structure Prior Dealing with ChIP-chip Data

Transcription factor binding data (ChiP-chip data) provides evidence of the existence of a regulatory relationship between a transcription factor and target genes in a genome. This evidence is often reported as *p*-values. Therefore, the probability of an edge being present in the true network is inversely related to this *p*-value. The smaller the *p*-value, the more likely the edge is to exist in the true network. To effectively integrate this type of high-throughput data into BN learning, Bernard et al. (2005) proposed the following model.

A *p*-value p_{ij} corresponding to an edge (*ij*) obtained from ChIP-chip data can be considered as an observation for random variable P_{ij} defined on the interval [0,1]. P_{ij} is assumed to be exponentially distributed if the edge (*ij*) is present in structure *S* and uniformly distributed if the edge (*ij*) is absent

from *S*. That is,

$$P_\lambda(P_{ij} = p_{ij} \mid (ij) \in S) = \frac{\lambda e^{-\lambda p_{ij}}}{1 - e^{-\lambda}}, \tag{10}$$

where λ is the parameter which controls the scale of the truncated exponential distribution, and $P(P_{ij} = p_{ij} \mid (ij) \notin S) = 1$. Further, let β denote the probability of the edge being present before observing the corresponding *p*-value. Then, by the Bayes rule, the probability that edge (*ij*) is present after observing the corresponding *p*-value is:

$$P_\lambda((ij) \in S \mid P_{ij} = p_{ij}) = \frac{\lambda e^{-\lambda p_{ij}} \beta}{\lambda e^{-\lambda p_{ij}} \beta + (1 - e^{-\lambda})(1 - \beta)}. \tag{11}$$

The mass of this distribution becomes more concentrated at smaller values of P_{ij} as the parameter λ increases. Conversely, the distribution spreads out and flattens as λ decreases. The *p*-value threshold can be determined by solving the equation $P_\lambda((ij) \in S \mid P_{ij} = p^*) = P_\lambda((ij) \notin S \mid P_{ij} = p^*)$, that is,

$$p_{ij}^* = \frac{-1}{\lambda} \log\left(\frac{(1 - e^{-\lambda})(1 - \beta)}{\lambda \beta}\right). \tag{12}$$

The relationship between λ and p^* can be explained as follows:

For any fixed value λ, an edge (*ij*) is more likely to be present than absent if the corresponding *p*-value is below this critical value p^*. As the value of λ increases, the value of p^* decreases. In this case we become more stringent about how low a *p*-value must be before we consider it as a prior evidence for an edge. Conversely, as λ decreases, the value of p^* increases. In this case, we become less stringent. As $\lambda \to 0$, it can be shown that $P_\lambda((ij) \in S \mid P_{ij} = p) \to \beta$, which is independent of *p*. This implies that if we have no confidence in the location data, the probability that edge (*ij*) is present is the same value β both before and after seeing the corresponding *p*-value. In other words, λ acts as a tuneable parameter indicating the degree of confidence in the evidence provided by the location data. This allows for modelling the noise level inherent in the location data.

The precise selection of λ could be difficult. To avoid the specification of a single value, the Bayesian approach can be adopted to compute a marginalized probability over λ. For convenience, it is assumed that λ is uniformly distributed over the interval $[\lambda_H - \lambda_L]$. Then let

$$P_\lambda((ij) \in S \mid P_{ij} = p_{ij}) = \frac{1}{\lambda_H - \lambda_L} \int_{\lambda_L}^{\lambda_H} \frac{\lambda e^{-\lambda p_{ij}} \beta}{\lambda e^{-\lambda p_{ij}} \beta + (1 - e^{-\lambda})(1 - \beta)} d\lambda. \tag{13}$$

The integral cannot be solved analytically, however, it can be solved numerically for a fixed p_{ij}. Since there are only finite many *p*-values for a set of location data, the integrals can be pre-computed.

The prior probability of a structure *S* usually uses the edge-wise decomposition:

$$P(S) = \prod_{(ij) \in S} P((ij) \in S \mid P_{ij} = p_{ij}) \prod_{(ij) \notin S} P((ij) \notin S \mid P_{ij} = p_{ij}). \tag{14}$$

In this formula, the term corresponding to the normalizing constant has been dropped.

Structure Prior from Multiple Sources of Biological Knowledge

Following the framework of Imoto et al. (2003) for the construction of structure prior, Werhi and Husmerier (2007) designed a scenario in which the energy takes on a particular form such that the computation of the marginal posterior distribution over the hyperparameter becomes analytically tractable. Furthermore, they extended the framework of Imoto et al. (2003) to include more than one energy function. This extended model allows for simultaneous inclusion of different sources of prior knowledge, such as promoter motifs and KEGG pathways. In their approach, all hyperparameters are sampled from the posterior distribution with Markov chain Monte Carlo (MCMC). The sampled hyperparameters ensure that the relative weights related to the different sources of prior knowledge are consistently inferred within the Bayesian context, while automatically trading off their relative influences in light of the data.

In that model, a matrix *B* that denotes the biological prior knowledge in which knowledge about interactions between nodes is represented by the entries $B_{ij} \in [0,1]$. More specifically,

$$B_{ij} = \begin{cases} 0.5 & \text{no prior knowledge about the presense or absence of} \\ & \text{an edge from node } i \text{ to node } j; \\ [0, 0.5) & \text{have prior evidence that there is no directed edge} \\ & \text{between node } i \text{ and node } j; \\ (0.5, 1] & \text{have prior evidence that there is a direct edge} \\ & \text{from node } i \text{ to node } j. \end{cases} \tag{15}$$

From the definitions of *B* and *E*(*S*), the energy of a structure *S* can be further defined as:

$$E(S) = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} | B_{ij} - s_{ij} |, \tag{16}$$

where s_{ij} is 1 if the edge (*ij*) is present in structure *S*, 0 otherwise. The energy is 0 for a perfect match between the prior knowledge *B* and the actual network structure *S*, while increasing with mismatches between *B* and *S*. Similar to those used in Imoto et al. (2003), the prior probability over network structures *S* is defined by taking the form of a Gibbs distribution:

$$P(S \mid \beta) = \frac{1}{Z(\beta)} e^{-\beta E(S)}, \text{ where } Z(\beta) = \sum_{S} e^{-\beta E(S)}. \tag{17}$$

The energy function *E*(*S*) can be further written as:

$$E(S) = \sum_{j=1}^{N} E(j, \pi_j) = \sum_{i \in \pi_j} (1 - B_{ij}) + \sum_{i \notin \pi_j} B_{ij}. \tag{18}$$

Then

$$Z(\beta) = \sum_{S} e^{-\beta E(S)} = \sum_{\pi_1} \cdots \sum_{\pi_N} e^{-\beta (E(1,\pi_1) + \cdots + E(N,\pi_N))} = \prod_{j=1}^{N} \sum_{\pi_j} e^{-\beta E(j,\pi_j)}. \tag{19}$$

Here, the summation is taken over all parent configurations π_j of node j in all network structures.

The above framework was further extended to multiple sources of prior knowledge (Werhli & Husmeier, 2007). Biological knowledge from each independent source is represented by a separate matrix B^k, $k = 1,\ldots, K$. Each matrix satisfies the requirements for B in formula (15). Therefore, K energy functions can be given as follows:

$$E_k(S) = \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \mid B_{ij}^k - s_{ij} \mid,\ k = 1,\ldots,K, \tag{20}$$

where each energy function is associated with its own hyperparameter β_k. The prior probability of a network structure S given the hyperparameter β_k can be defined as:

$$P(S \mid \beta_1,\ldots,\beta_K) = \frac{1}{Z(\beta_{1,\ldots,}\beta_K)} e^{-(\beta_1 E_1(S)+\bullet\bullet\bullet+\beta_K E_K(S))}, \tag{21}$$

where the partition function is given by

$$Z(\beta_1,\ldots\beta_k) = \sum_S e^{-(\beta_1 E_1(S)+\bullet\bullet\bullet+\beta_K E_K(S))}. \tag{22}$$

Similarly,

$$E_k(S) = \sum_{j=1}^{N} E_k(j,\pi_j) = \sum_{i\in\pi_j}(1 - B_{ij}^k) + \sum_{i\notin\pi_n} B_{ij}^k, \tag{23}$$

$$\begin{aligned} Z(\beta) = \sum_S e^{-(\beta_1 E_1(S)+\cdots+\beta_k E_K(S))} &= \sum_{\pi_1}\ldots\sum_{\pi_N} e^{-(\beta_1(E_1(1,\pi_1)+\bullet\bullet\bullet+E_1(N,\pi_N))+\bullet\bullet\bullet+\beta_K(E_k(1,\pi_1)+\bullet\bullet\bullet+E_K(N,\pi_N)))} \\ &= \prod_{j=1}^{N}\sum_{\pi_j} e^{-(\beta_1 E_1(j,\pi_j)+\bullet\bullet\bullet+\beta_K E_K(j,\pi_j))} \end{aligned} \tag{24}$$

The MCMC scheme can be used to sample both network structures and hyperparameters from the posterior distributions. The details on the general MCMC procedure can be found in Chapter 'Bayesian Networks for Modelling and Inferring Gene Regulatory Networks'. In their simulations, they chose the prior distribution of the hyperparameters $P(\beta)$ to be the uniform distribution over the interval [0, *MAX*]. The proposal probability for the hyperparameters $R(\beta_{new} \mid \beta_{old})$ was chosen to be a uniform distribution over a moving interval of length $l \leq\leq MAX$, centered on the current value of the hyperparameter.

Structure Prior Dealing with Literature

Almasri et al. (2008) proposed a structure prior for gene interactions using LOI-scores obtained through the statistical analysis of interactions in literature (Larsen, Almasri, Chen & Dai, 2007). The confidence in a possible interaction between a pair of genes is measured by the *p*-value of the LOI-score with the assumption that the LOI-score for the gene pair follows a normal distribution. If the *p*-value of an LOI-score is significant, then the corresponding interaction is believed to be more likely. Conversely, if the *p*-value of an LOI-score is insignificant, then belief that the corresponding gene pairs interact should be

lower. The detailed assignment of prior probability for gene pairs is described as below.

The structure prior for the edge from gene *i* to gene *j* is then assigned as:

$$\pi_{ij} = p(i \rightarrow j) = 1\text{-}\ p_{ij},\ p(i \bullet\bullet\bullet j) = 1 - \pi_{ij} = p_{ij}, \tag{25}$$

where and $i \bullet\bullet\bullet j$ means that there is an edge or there is no edge from gene *i* to gene *j*, respectively. Let e_{ij} denote the random variable that takes value 1 if there is an edge from gene *i* to gene *j* and takes value 0 otherwise. Then, from the Bernoulli distribution the probability for random variable e_{ij} is:

$$p(e_{ij}) = \pi_{ij}^{e_{ij}}(1-\pi_{ij})^{1-e_{ij}}. \tag{26}$$

The structure prior constructed in this way is only an informal prior. A formal prior for the BN structure *S* can be written as follows:

$$P(S) = c\prod_{(ij)\in S} p(e_{ij})\prod_{(ij)\notin S} p(e_{ij}) \tag{27}$$

where *c* is a normalizing constant. The normalizing constant *c* can be fixed at 1, as the actual magnitude of *c* does not affect structure searching (Castelo & Siebes, 1998).

Structure Prior Dealing with Errors in Literature Knowledge

Imoto et al. (2006) proposed a model that can handle errors in literature or errors accumulated during the process of data collection. Given that the prior knowledge about the regulative interactions among some genes is stored in biological databases, this information is recorded in a matrix B^0 with entries defined as $B_{ij}^0 = 1$ if it is known that gene *i* regulates gene *j*; $B_{ij}^0 = 2$ if it is known that gene *i* does not regulate gene *j*. $B_{ij}^0 = 0$ if nothing is known about gene *i* and gene *j*. As commented in Imoto et al. (2006), the negative information such as gene *i* does not regulate gene *j*, i.e., $B_{ij}^0 = 2$, usually is not included in the database. However, using additional information such as subcellular localization can create the negative set.

Their model is to find the optimal network $\hat{S}$ and the optimal updated database information $\hat{B}$ that maximizes the conditional joint probability:

$$P(S, B \mid D, B^0), \tag{28}$$

where each entry in matrix *B* is the updated information from B^0 and *D* is the gene expression data. The conditional joint probability is then rewritten as

$$P(S, B \mid D, B^0) = \frac{P(D, S, B \mid B^0)}{P(D \mid B^0)}, \tag{29}$$

where

$$P(D \mid B^0) = \sum_S \sum_B P(S, B, D \mid B^0) \tag{30}$$

is the normalizing constant and does not relate to the selection of *S* and *B*. Therefore, given *D*, the maximization of the conditional joint probability $P(S, B|D, B^0)$ is equivalent to the maximization of $P(D,S, B|B^0)$. When the database information B^0 is given, the conditional joint probability can be decomposed as

$$P(D, S, B|B^0) = P(D|S)P(S|B)P(B|B^0). \tag{31}$$

Since B_{ij} can take one of the values 0, 1, or 2, we let $\beta_0, \beta_1, \beta_2$ be the parameters associated with each value respectively. Further, set $\beta_0 = 0 < \beta_1 < \beta_2$. Similar to the model described in the previous section, the prior probability of the graph *S* can be expressed as

$$P(S \mid B) = \frac{1}{Z} \sum_{(ij)\in S} e^{-\beta_{B_{ij}}} \tag{32}$$

The parameters β_1, β_2 need to be optimized. The computation of normalizing constant is intractable even for moderately sized gene networks. However, it is possible to obtain the exact value of *Z* for the dynamic Bayesian network models. The details can be found in Imoto et al. (2006).

The conditional probability represents the transition probability when we update the database information from B^0 to *B*. The statistical model for $P(B|B^0)$ is essential to realize a self-repairing system for biological database.

First, define a function $d(a)$ to categorize edges into two groups:

$$d(a) = \begin{cases} 1 \text{ for } a = 1 \text{ or } 2, \\ 0 \text{ for } a = 0. \end{cases} \tag{33}$$

Then transition probability $P(d(B_{ij}) \mid d(B_{ij}^0))$ is then constructed by using the Bernoulli distribution of the form

$$P(d(B_{ij}) \mid d(B_{ij}^0)) = P(d(B_{ij}^0))^{d(B_{ij})})[1 - P(d(B_{ij}^0))]^{1-d(B_{ij})}. \tag{34}$$

Then $P(B|B^0)$ can be modeled by the product of the Bernoulli distributions

$$P(B \mid B^0) = \prod_{i=1}^{n} \prod_{j=1}^{n} P(d(B_{ij}^0))^{d(B_{ij})} [1 - P(d(B_{ij}^0))]^{1-d(B_{ij})}. \tag{35}$$

They set a high probability for $P(d(B_{ij}^0) = 1),$ because the information on edges with corresponding to $B_{ij}^0 = 1 \text{ or } 2$ is rather reliable. One the other hand, since there is no information about edges with $B_{ij}^0 = 0$, it is reasonable to set the probability $P(d(B_{ij}^0) = 1),\ B_{ij}^0 = 1 \text{ or } 2\ B_{ij}^0 = 0\ \ P(d(B_{ij}^0) = 0) = 0.5.$ because the information on edges with corresponding to $B_{ij}^0 = 1 \text{ or } 2$ is rather reliable. One the other hand, since there is no information about edges with $B_{ij}^0 = 0$, it is reasonable to set the probability $P(d(B_{ij}^0) = 0) = 0.5.$ In addition, if the edge from gene *i* to gene *j* is stored as a known relationship in the database, but the edge is not observed from the gene expression data, then the edge is removed from the database by setting $B_{ij} = 0$ if it leads to an increase in the conditional joint probability $P(D, S, B|B^0) = P(D|S)P(S|B)P(B|B^0)$. Conversely, if the edge from gene *i* to gene *j* is clearly observed from the gene

expression data, but the edge is not included in the database, this edge is added to the database by setting $B_{ij} = 1$ if the conditional joint probability increases.

COMPUTATIONAL EXPERIMENTS

The networks used in computational experiments with previously described structure priors are of relatively small scale, ranging from 25 to 125 genes (Almasri, Larsen, Chen & Dai, 2008; Le, Bahl & Unga, 2004; Werhli & Husmeier, 2007). The accurate quantification of the improvement when prior biological knowledge is incorporated into a BN framework is difficult due to lack of knowledge of the true biological network structure. We highlight several experimental studies here.

A genetic network involved in hepatic glucose homeostasis was used to generate a synthetic microarray data (Le, Bahl & Unga, 2004). Then the ability of the BN approach to reconstruct networks and reduce the amount of data required was analyzed when different types of prior biological knowledge were incorporated. The considered network has 35 nodes and 53 interactions. Prior knowledge was used in a constraint-based approach in the following two ways: (1) add edges that are known to regulate a target gene, and (2) remove edges that are known not to regulate a target gene. It was reported that the number of expression profiles required to learn a network with fixed level of sensitive is less when type (1) edges were added. It was also shown there is a general increase in sensitivity when more of type (2) edges were removed.

The genetic network used in Almasri et al. (2008) has 102 genes and 171 interactions that are involved in transcription regulation related to the yeast cell cycle. The microarray data are a time series gene expression that covers more than two complete cycles of the cell division (Spellman, Sherlock, Zhang & Iyer, 1998). It is highly enriched for known interacting genes involved in the *Saccharomyces* cell cycle. In their study two ways of incorporating prior knowledge were investigated: (a) use a prior structure prior defined from formulas (27), and (b) use prior knowledge to restrict the search. In (b) an interaction is considered a possible candidate if its LOI score is greater than certain threshold and otherwise is considered impossible and removed. Both approaches were able to generate networks with improved quality in terms of sensitivity, precision and biological relevance. In addition, the second approach seems to have slightly better performance.

In their work (Werhli & Husmeier, 2007), the computational study revealed that prior knowledge that is more consistent with the data is given a stronger weight by the Bayesian inference scheme. The study also provided the evidence that the proposed Bayesian inference method can discriminate between different sources of prior knowledge and automatically assess their relative merits through learning the hyperparameters associated with the prior knowledge. The influence of an irrelevant prior will be automatically suppressed; however, the prior will not be completely switched off. They also quantified the qualities of learned networks using and without using the prior knowledge with several criteria: (1) the number of learned true undirected edges, (2) the number of learned true edges, and (3) the area under the receiver operator characteristics (ROC) curve, where the relative number of true positive edges is plotted against the relative number of the false positive edges. It was clear that the method using prior knowledge outperformed the ones that do not use all evaluation criteria. There are also some controversial results. It was shown in (Geier, Timmer & Fleck, 2007) that the benefit of using prior knowledge is limited to conditions of small time series gene expression data. Their prior knowledge is given by

the probability distributions for true and false interactions respectively. Another issue is that the prior interaction probabilities are drawn from two truncated normal distributions. The accuracy of the prior knowledge is controlled by a parameter that is used to separate the means of both distributions. It is not clear if this assumption can be satisfied by prior biological knowledge.

Gevaert et al. (2007) investigated two complementary sources of information: PubMed abstracts combined with publicly available taxonomies or ontologies, and known protein–DNA interactions. These priors, either separately or combined, have the potential to reduce the complexity of learning reverse-engineering regulatory networks while creating more robust and reliable models. Moreover, this approach can easily be extended with other data sources.

CONCLUSION

Incorporation of prior biological knowledge into the BN learning framework is crucial for successful inference of gene networks from microarray gene expression data. Bayesian networks provide a powerful framework for data integration and regulatory network modeling. We have presented several methods for constructing structure priors from various types of knowledge, including curated databases, high-throughput experimental data, literature, and computational analysis results from sequences information other than experiments. However, lack of a systematic evaluation of the performance presents a serious problem in the current research.

FUTURE RESEARCH DIRECTIONS

It is important to understand the relative merits and shortcomings for various proposed structure priors. The comparison is not a simple task. Stolovitzky et al. (2007) pointed out that the needs and challenges for the establishment of a set of protocols that can achieve a fair comparison of the strengths and weaknesses of the inference methods and a clear sense of the reliability of the network models produced. The Dialogue for Reverse Engineering Assessments and Methods (DREAM) (http://wiki.c2b2.columbia.edu/dream/index.php/The_DREAM_Project) is set up for such a need. The second DREAM meeting has posted challenging problems. The study of effectively incorporating prior structures could benefit by using the appropriate challenging problems. Some networks used in recent studies of network algorithm comparison could also be used for evaluating structure prior in BN method (Soranzo, Bianconi & Altafini, 2007; Werhli, Grzegorczyk & Husmeier, 2006).

REFERENCES

Akutsu, T., Miyano, S., & Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing* (pp. 17-28).

Almasri, E., Larsen, P., Chen, G., & Dai, Y. (2008). Incorporating literature knowledge in Bayesian network for inferring gene networks with gene expression data. *4th International Symposium on Bioinformatics Research and Applications* (pp. 184-195). Springer-Verlag.

Ashburner, M., Ball, C. A., Blake, J. A., & Botstein, D. (2000). Gene ontology: Tool for the unification of biology. The gene ontology consortium. *Nature Genetics*, *25*, 25–29. doi:10.1038/75556

Baeza-Yates, R., & Ribeiro, B. A. N. (1999). *Modern information retrieval*. New York: Addison-Wesley.

Battle, A., Segal, E., & Koller, D. (2005). Probabilistic discovery of overlapping cellular processes and their regulation. *Journal of Computational Biology*, *12*(7), 909–927. doi:10.1089/cmb.2005.12.909

Beinlich, I. A., Suermondt, H., Chavez, R., & Cooper, G. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *Second European Conference in Artificial Intelligence in Medicine* (pp. 247-256).

Bernard, A., & Hartemink, A. J. (2005). Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. *Pacific Symposium on Biocomputing* (pp. 459-470).

Campos, L. M. d., & Castellano, J. G. (2007). Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning*, *45*(2), 233–254. doi:10.1016/j.ijar.2006.06.009

Castelo, R., & Siebes, A. (1998). Priors on network structures. Biasing the search for Bayesian networks (Tech. Rep. INS-R9816). CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands.

Castelo, R., & Siebes, A. (2000). Priors on network structures. Biasing the search for Bayesian networks. *International Journal of Approximate Reasoning*, *24*(1), 39–57. doi:10.1016/S0888-613X(99)00041-9

Cheng, J., Bell, D. A., & Liu, W. (1997). Learning belief networks from data: An information theory based approach. *The Sixth ACM International Conference on Information and Knowledge Management* (pp. 325-331).

Chickering, D. M. (2002). Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, *2*, 445–498. doi:10.1162/153244302760200696

Chien, C., Bartel, P. L., Sternglanz, R., & Fields, S. (1991). The two-hybrid system: A method to identify and clone genes for proteins that interact with a protein of interest. *Proceedings of the National Academy of Sciences of the United States of America*, *88*(21), 9578–9582. doi:10.1073/pnas.88.21.9578

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*(4), 309–347.

Cooper, G. F. (1997). A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, *1*(2), 203–224. doi:10.1023/A:1009787925236

Elnitski, L., Jin, V. X., Farnham, P. J., & Jones, S. J. M. (2006). Locating mammalian transcription factor binding sites: A survey of computational and experimental techniques. *Genome Research*, *16*(12), 1455–1464. doi:10.1101/gr.4140006

Friedman, N., & Koller, D. (2003). Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, *50*(1-2), 95–125. doi:10.1023/A:1020249912095

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3-4), 601–620. doi:10.1089/106652700750050961

Gavin, A.-C., Bosche, M., Krause, R., & Grandi, P. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, *415*(6868), 141–147. doi:10.1038/415141a

Geier, F., Timmer, J., & Fleck, C. (2007). Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge. *BMC Systems Biology*, *1*(1), 11. doi:10.1186/1752-0509-1-11

Gevaert, O., Van Vooren, S., & De Moor, B. (2007). A framework for elucidating regulatory networks based on prior information and expression data. *Annals of the New York Academy of Sciences*, *1115*(1), 240–248. doi:10.1196/annals.1407.002

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing* (pp. 422-33).

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197–243.

Heckerman, D., Meek, C., & Cooper, G. (2006). A Bayesian approach to causal discovery. *Innovations in Machine Learning*, 1.

Ho, Y., Gruhler, A., Heilbut, A., & Bader, G. D. (2002). Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, *415*(6868), 180. doi:10.1038/415180a

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics (Oxford, England)*, *19*(17), 2271–2282. doi:10.1093/bioinformatics/btg313

Iliopoulos, I., Enright, A. J., & Ouzounis, C. A. (2001). Textquest: Document clustering of Medline abstracts for concept discovery in molecular biology. *Pacific Symposium on Biocomputing* (pp. 384-395).

Imoto, S., Goto, T., & Miyano, S. (2002). Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. *Pacific Symposium on Biocomputing* (pp. 175-186).

Imoto, S., Higuchi, T., Goto, T., & Miyano, S. (2006). Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. *Statistical Methodology*, *3*(1), 1–16. doi:10.1016/j.stamet.2005.09.013

Imoto, S., Higuchi, T., Goto, T., & Tashiro, K. (2003). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *The IEEE Computer Society Conference on Bioinformatics* (pp. 104-113). IEEE Computer Society.

Imoto, S., Sunyong, K., Goto, T., & Aburatani, S. (2002). Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *The IEEE Computer Society Conference on Bioinformatics* (pp.219-227). IEEE Computer Society.

Ito, T., Ota, K., Kubota, H., & Yamaguchi, Y. (2002). Roles for the two-hybrid system in exploration of the yeast protein interactome. *Molecular & Cellular Proteomics*, *1*(8), 561–566. doi:10.1074/mcp.R200005-MCP200

Larraaga, P., Poza, M., Yurramendi, Y., & Murga, R. H. (1996). Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(9), 912–926. doi:10.1109/34.537345

Larsen, P., Almasri, E., Chen, G., & Dai, Y. (2007). A statistical method to incorporate biological knowledge for generating testable novel gene regulatory interactions from microarray experiments. *BMC Bioinformatics*, *8*, 317. doi:10.1186/1471-2105-8-317

Le, P. P., Bahl, A., & Unga, L. H. (2004). Using prior knowledge to improve genetic network reconstruction from microarray data. *In Silico Biology, 4*, 0027.

Lee, T. I., Rinaldi, N. J., Robert, F., & Odom, D. T. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, *298*(5594), 799–804. doi:10.1126/science.1075090

Nariai, N., Kim, S., Imoto, S., & Miyano, S. (2004). Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 336–347.

Neapolitan, R. E. (2003). *Learning Bayesian networks*. Prentice Hall.

Pe'er, D., Regev, A., Elidan, G., & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics (Oxford, England)*, *17*(Suppl 1), S215–S224.

Shoemaker, B. A., & Panchenko, A. R. (2007). Deciphering protein-protein interactions. Part I. Experimental techniques and databases. *PLoS Computational Biology*, *3*(3), e42. doi:10.1371/journal.pcbi.0030042

Singh, M., & Valtorta, M. (1993). An algorithm for the construction of (Bayesian) network structures from data. *The Ninth Conference on Uncertainty in Artificial Intelligence* (pp. 259-265). Washington, D.C.: Morgan Kaufmann.

Soranzo, N., Bianconi, G., & Altafini, C. (2007). Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: Synthetic vs. real data. *Bioinformatics (Oxford, England)*, *23*(13), 1640–1647. doi:10.1093/bioinformatics/btm163

Spellman, P. T., Sherlock, G., Zhang, M. Q., & Iyer, V. R. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, *9*(12), 3273–3297.

Spirtes, P., Glymour, C. N., & Scheines, R. (1993). *Causation, prediction, and search*. New York: Springer-Verlag.

Tamada, Y., Bannai, H., Imoto, S., & Katayama, T. (2005). Utilizing evolutionary information and gene expression data for extimating gene networks with Bayesian network models. *Journal of Bioinformatics and Computational Biology*, *3*(6), 1295–1313. doi:10.1142/S0219720005001569

Tamada, Y., Kim, S., Bannai, H., & Imoto, S. (2003). Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics (Oxford, England)*, *19*(Suppl 2), II227–II236. doi:10.1093/bioinformatics/btg1082

Uetz, P., Giot, L., Cagney, G., & Mansfield, T. A. (2000). A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, *403*, 623–627. doi:10.1038/35001009

Van den Bulcke, T., Lemmens, K., Van de Peer, Y., & Marchal, K. (2006). Inferring transcriptional networks by mining omics data. *Current Bioinformatics*, *1*, 301. doi:10.2174/157489306777827991

Wang, M., Chen, Z., & Cloutier, S. (2007). A hybrid Bayesian network learning method for constructing gene networks. *Computational Biology and Chemistry*, *31*(5-6), 361–372. doi:10.1016/j.compbiolchem.2007.08.005

Werhli, A. V., Grzegorczyk, M., & Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics (Oxford, England)*, *22*(20), 2523–2531. doi:10.1093/bioinformatics/btl391

Werhli, A. V., & Husmeier, D. (2007). Reconstructing gene regulatory betworks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, *6*(1). doi:10.2202/1544-6115.1282

Whitfield, M. L., George, L. K., Grant, G. D., & Perou, C. M. (2006). Common markers of proliferation. *Nature Reviews. Cancer*, *6*(2), 99. doi:10.1038/nrc1802

Wray, B. (1991). Theory refinement on Bayesian networks. *The Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 52-60). Los Angeles: Morgan Kaufmann Publishers Inc.

KEY TERMS AND DEFINITIONS

Gene Ontology: The Gene Ontology (GO) project has developed three structured controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner.

K2 Algorithm: K2 algorithm is a score-based algorithm in Bayesian network. It recovers the underlying graphic structure based on a predetermined order of nodes in a greedy fashion.

KEGG Pathways: Kyoto Encyclopedia of Genes and Genomes (KEGG) includes a collection of manually drawn pathway maps representing our knowledge on the molecular interaction and reaction networks for metabolism, genetic information processing, environmental information processing, cellular processes, human diseases and drug development.

LOI-Score: The likelihood of interaction (LOI) score is a measure of the likelihood that a gene or a gene product with a particular molecular function interacts with another gene or a gene product of a particular molecular function. These scores can be derived from databases of interactions and the Gene Ontology molecular function annotations.

Prior Biological Knowledge: Databases that include biological interactions between proteins, proteins and DNAs. These interactions are collected from high throughput experiments or curated from literature.

Structural Prior: Structure prior is a prior distribution over directed acyclic graphs.

Chapter 13
Problems for Structure Learning:
Aggregation and Computational Complexity

Frank Wimberly
Carnegie Mellon University (retired), USA

David Danks
Carnegie Mellon University and Institute for Human & Machine Cognition, USA

Clark Glymour
Carnegie Mellon University and Institute for Human & Machine Cognition, USA

Tianjiao Chu
University of Pittsburgh, USA

ABSTRACT

Machine learning methods to find graphical models of genetic regulatory networks from cDNA microarray data have become increasingly popular in recent years. We provide three reasons to question the reliability of such methods: (1) a major theoretical challenge to any method using conditional independence relations; (2) a simulation study using realistic data that confirms the importance of the theoretical challenge; and (3) an analysis of the computational complexity of algorithms that avoid this theoretical challenge. We have no proof that one cannot possibly learn the structure of a genetic regulatory network from microarray data alone, nor do we think that such a proof is likely. However, the combination of (i) fundamental challenges from theory, (ii) practical evidence that those challenges arise in realistic data, and (iii) the difficulty of avoiding those challenges leads us to conclude that it is unlikely that current microarray technology will ever be successfully applied to this structure learning problem.

DOI: 10.4018/978-1-60566-685-3.ch013

INTRODUCTION

An important goal of cell biology is to understand the network of dependencies through which genes in a tissue type regulate the synthesis and concentrations of protein species. A mediating step in such synthesis is the production of messenger RNA (mRNA). Protein products of one gene may help to regulate the rate of transcription into mRNA of the DNA reading frame of certain other genes. These dependencies among gene activities and their mRNA proxies have long been represented by directed graphs. Early in the 1990s, machine learning algorithms were developed for learning directed graphs representing causal relations from appropriate data samples. At about the same time, developments in microarray techniques made possible the simultaneous measurement of messenger RNA (mRNA) counts for thousands of distinct genes. This juxtaposition naturally led to a flood of studies in the computer science and biological literatures applying various search algorithms to gene expression data, with the aim of producing directed graphs that describe, for a tissue type, which genes regulate transcription rates of which other genes. Some of that work continues. We now know that the machine learning techniques are inappropriate and unsound in these applications, although they are potentially applicable to more recent measurements of RNA transcript concentrations in single cells. This chapter explains the statistical reasons why, as well as some of the relevant issues of computational complexity.

The short story is this: The goal of inference is the regulatory network within individual cells, but current microarray measurements are of mRNA counts extracted from large samples of cells. The machine learning algorithms exploit assumed symmetries between the network structure and a class of statistical properties of measurements. Assuming those symmetries hold for mRNA concentrations in individual cells and the regulatory network in the individual cellular level, and assuming all cells in the measured sample have the same regulatory network, it follows that the symmetry fails for measurements of concentrations aggregated from multiple cells. Experimental studies with real and simulated data confirm this failure.

THEORY: LEARNING FROM AGGREGATIONS

Microarrays are small chips a few square inches in size on which spots of DNA have been imbedded. A typical chip may contain thousands of spots, each spot composed of multiple copies of a small sequence of DNA. In the living cell nucleus, sections of DNA are copied ("transcribed") into a dual complementary molecule, RNA, which is the scaffolding for the synthesis, outside the cell nucleus, of cellular proteins. RNA can be extracted from tissue, and tiny luminescent beads can be chemically attached to RNA molecules obtained from tissue cells (e.g., from breast cancer cells). Each RNA molecule contains a sequence of bases that binds to a specific DNA sequence. When a suspension consisting of many RNA molecules from a tissue sample is applied to a microarray, the RNA molecules bind to the complementary DNA sites. By measuring the luminosity of each DNA spot, the relative concentration of each kind of RNA in the tissue sample can be estimated. From these concentrations, one can infer relative activity of genes—how much RNA is produced by various parts of the cell DNA in the tissues sampled.

Two fundamentally different strategies have been proposed to determine networks of regulatory relationships from microarray measurements. One strategy (Davidson, *et al*., 2002; Ideker, *et al*., 2001; Yuh, Bolouri, & Davidson, 1998) experimentally suppresses (or enhances) the expression of one or more genes, and measures the resulting increased or decreased expression of other genes. The method,

while laborious, has proved fruitful in unraveling small pieces of the regulatory networks of several species. Its chief disadvantage is that each experiment provides information only about the effects of the manipulated gene or genes. A single knockout of gene *A* resulting in changed expression of genes *B* and *C*, for example, does not of itself provide information as to whether *A* regulates both *B* and *C* directly, or whether *A* regulates *B* which in turn regulates *C*, or whether *A* regulates *C* which in turn regulates *B*. And if manipulation of *A* yields no association with *B*, then *B* may still influence *A* (or not). This implies that at least *N*–1 experiments that intervene on a single gene would be required to identify the dependency structure of *N* genes, assuming those genes have no additional, unmeasured common causes, and this laborious procedure still cannot distinguish between direct and indirect regulation. Statistically, one uses only the estimation of the expression level of each gene considered in each experiment, and the uncertainties of those estimates. Experiments with multiple simultaneous interventions on gene expression complicate matters in ways we will discuss later.

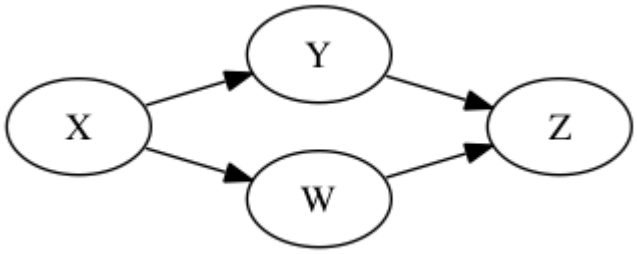
A second strategy relies on the natural variation of expression levels of the same gene in different cells. The proposal is to measure—typically with microarrays—the expression levels in repeated samples from the same tissue source, or similar sources, and to infer the regulatory structure from the statistical dependencies and independencies among the measured expression levels (Akutsu, Miyano, & Kuhara, 1998; D'haeseleer, 2000; D'haeseleer, Liang, & Somogyi, 2000; Friedman, Nachman, & Pe'er, 2000; Hartemink, 2001; Hashimoto *et al*., 2004; Liang, Fuhrman, & Somogyi, 1998; Shmulevich, Dougherty, Kim, & Zhang, 2002; Shrager, Langley, & Pohorille, 2002; Yoo, Thorsson, & Cooper, 2002). The apparent advantage of the strategy is that it offers the possibility of determining multiple relationships without separate experimental interventions. If, for example, gene *A* regulates gene *C* only by regulating gene *B* which in turn regulates *C*, the expression level of *A* should be independent, or nearly independent, of the expression level of gene *C* conditional on the expression level of gene *B*. In principle, if adequate sample sizes were available, the methods could also be used as a supplement to gain additional information from experiments in which the expression of particular genes are experimentally suppressed or enhanced (but see the Computational Complexity section below). The requisite statistical procedures for this strategy are more elaborate, and require direct or indirect (e.g. implicit in the posterior probabilities) estimates of conditional independence relationships among expression levels.

There are many statistical obstacles to the second strategy including: the joint influence of unmeasured factors (e.g. unmeasured gene expressions or extra-cellular factors), a variety of sources of measurement error, an unknown family of probability distributions governing the errors, and functional dependencies for the expression of any gene that may be Boolean for some regulating genes and continuous for other regulators. Some of these difficulties—in particular the presence of unrecorded common causes—can, in principle, be overcome (Spirtes, Glymour, & Scheines, 2001). We describe in this section a more elementary statistical difficulty with the second strategy that calls its value into question and raises a set of important research problems.

Directed Acyclic Graphs and Markov Factorization

Qualitative regulatory relationships among genes are often represented by directed graphs. Each vertex is a random variable whose values represent levels of expression of a particular gene. Each directed edge from a variable *X* to a variable *Y* in such a graph indicates that *X* produces a protein that regulates *Y*. In principle, the graph may be cyclic or acyclic, and may even have self-loops (a directed edge from a vari-

Figure 1. Example directed acyclic graph



able to itself). In the simplest case, one assumes an acyclic graph with noises and random measurement errors for each measurement of each gene that are independent of those for any other gene.

We consider this simplest case: the true, but unknown regulatory structure can be represented by a directed acyclic graph, with independent errors. Consider, for example, four genes *X*, *Y*, *Z*, *W* whose regulatory connections can be represented by Figure 1.

Suppose the measured values of *X*, *Y*, *Z*, *W* satisfy the following three equations:

$$Z = f(Y,W) + \varepsilon_Z$$

$$Y = g(X) + \varepsilon_Y$$

$$W = h(X) + \varepsilon_W$$

f, *g*, *h* are any functions and ε_Z, ε_Y, ε_W are independently distributed noises. It follows that the joint probability density of *X*, *Y*, *Z*, *W* admits a Markov factorization: $d(X, Y, Z, W) = d(Z \mid Y, W)\, d(Y \mid X)\, d(W \mid X)\, d(X)$. The Markov factorization implies that *Y* and *W* are independent conditional on *X*, and that *X* and *Z* are independent conditional on {*Y*, *W*}; it is in fact equivalent to specifying that these two relationships hold. More generally, assuming each random variable has an independent noise source but is otherwise a deterministic function of its parents in the graph, the system described by any directed acyclic graph has a density that admits a Markov factorization that can be written as the product, over all variables, of the density of each variable conditional on its graphical parents. Graphs with the same Markov factorization imply the same independencies and conditional independencies, and so form an equivalence class. The Markov equivalence class for Figure 1 consists of that graph and the graphs obtained by reorienting exactly one of: $X \rightarrow Y$ or $X \rightarrow W$. The Markov equivalence class represents the most information that could be obtained from second moments of the joint distribution of the variables. Non-Normal distributions have higher moments that are not uniquely determined by the second moment, and for linear systems it has been shown that higher moments can resolve structure more finely than the Markov equivalence class. We focus here on the dominant type of search algorithm for gene regulation networks: namely, those that assume either linearity or rely exclusively on second moments, even though some independently established expression dependencies are known to be non-linear. For time series data, regulatory relationships can still be represented by a directed acyclic graph and probabilities admitting a Markov factorization, but with vertices appropriately labeled by gene and time.

The Challenge of Aggregation

In structure learning, the aim is to discover the regulatory structure in individual cells, but measurements are typically of relative concentrations of mRNA transcripts obtained from thousands, or even millions, of cells. Such measurements are not of variables such as X in Figure 1, but are instead, ideally, of the sum of the X values over many cells. We will denote such measured sums over n cells by ΣX_i.

In general, the conditional dependencies/independencies among the gene expression levels of a single cell are not the same as those among the sums of gene expression levels over a number of cells. This statistical fact poses a serious difficulty for the second strategy for regulatory structure inference, which relies on the statistical dependencies among the gene expression levels. For example, if the variables in Figure 1 are binary, and each measurement is of the aggregate of transcript concentrations from two or more cells, ΣX_i and ΣZ_i are not independent conditional on $\{\Sigma Y_i, \Sigma W_i\}$, and the associations obtained from repeated samples will not therefore satisfy the Markov factorization (Danks & Glymour, 2002).

There are some special cases where the conditional independencies are invariant under aggregation. For example, if binary regulatory relations among genes X, Y, and Z are described by a singly connected graph such as $X \rightarrow Y \rightarrow Z$, $X \leftarrow Y \leftarrow Z$, or $X \leftarrow Y \rightarrow Z$, then the implied conditional independence of X and Z given Y holds as well for sums of independent measurements of X, Y, and Z respectively (Danks & Glymour, 2002).

Linear, Normal distributions have special virtues for invariance. Whatever the directed acyclic graph of cellular regulation may be, if each variable is a linear function of its parents and an independent Gaussian noise, then the Markov factorization holds for the summed variables. In that case, conditional independence is equivalent to vanishing partial correlation, and the partial correlation of the two variables (although not the sampling distribution), each respectively composed of the sum of n like variables, will be the same as the partial correlation of the unsummed variables.

Two less restrictive sufficient conditions for conditional independence of variables to be the same as the conditional independence of their sums, are given in two theorems proved in Chu, Glymour, Scheines, & Spirtes (2003):

Theorem 1 (Local Markov theorem): Given a directed acyclic graph G representing the causal relations among a set $\mathbf{V}$ of random variables, let $Y, X_1, \ldots, X_k \in \mathbf{V}$, and $\mathbf{X} = \{X_1, \ldots, X_k\}$ be the parents of Y in G. If $Y = \mathbf{c}^T\mathbf{X} + \varepsilon$, where $\mathbf{c}^T = (c_1, \ldots, c_k)$, and ε is a noise term independent of all non-descendents of Y, then Y is independent of all its non-parents, non-descendents conditional on its parents $\mathbf{X}$, and this relation holds under aggregation.

Theorem 2 (Markov wall theorem): Given a directed acyclic graph G representing the causal relations among a set $\mathbf{V}$ of random variables. Let $\mathbf{X} = \{X_1, \ldots, X_h\}$, $\mathbf{Y} = \{Y_1, \ldots, Y_k\}$, $\mathbf{W} = \{W_1, \ldots, W_m\}$, and $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{W} = \mathbf{V}$. Suppose that the following three conditions hold:

1. The joint distribution of $X_1, \ldots, X_h, Y_1, \ldots, Y_k$ is multivariate normal with non-singular covariance matrix;
2. For $i = 1, \ldots, k$, Y_i is neither a parent nor a child of any variable $W_j \in \mathbf{W}$ (i.e., there is no direct edge between a variable in $\mathbf{Y}$ and a variable in $\mathbf{W}$); and
3. For $i = 1, \ldots, h$, X_i is not a child of any variable $W_j \in \mathbf{W}$ (i.e., any edge between a variable in $\mathbf{X}$ and a variable in $\mathbf{W}$ must be from the $\mathbf{X}$-variable to the $\mathbf{W}$-variable).

Then **Y** is independent of **W** conditional on **X**, and this relation holds under aggregation.

Although there are established regulatory mechanisms in which some regulators of a gene act linearly in the presence of a suitable combination of other regulators of the same gene (Yuh, *et al*., 1998), there does not appear to be any known regulatory system that is simply linear.

One of the best-established regulatory functional relations seems to be the expression of the Endo16 gene of the sea urchin (Yuh, *et al*., 1998). The expression level of the gene is controlled by a Boolean regulatory switch between two functions, each of which is a product of a Boolean function of regulator inputs multiplied by a linear function of other regulator inputs. Even much simplified versions of such transmission functions do not preserve conditional independence over sums of variables.

Consider an example: suppose in each of n cells genes X, Y, Z, and W have the regulatory structure $X \rightarrow Y \rightarrow Z \leftarrow W$ with $Y = X^2$; $Z = YW$; and W a binary variable with $P(W = 1) = p$. Assume X takes values in {0, 1, 2, 3, 4} with uniform probability. Let ΣX_i, ΣY_i, ΣZ_i, and ΣW_i denote the sums of values of X, Y, Z and W respectively over $n = 4$ cells. Z is independent of X given Y in each cell. However, we will show that ΣZ_i is not independent of ΣX_i given ΣY_i.

For each cell i, $Z_i = Y_i$ if the value of W_i is 1, and zero otherwise. Hence the probability that $Z_i = y_i$ given that $Y_i = y_i$ is p+(1-p)/5. Let $\Sigma Y_i = \Sigma((X_i)^2 = 16$. There are just five possible vector values for $\mathbf{X} = \langle X_1, X_2, X_3, X_4\rangle$ consistent with $\Sigma((X_i)^2 = 16$: <4, 0, 0, 0>; <0, 4, 0, 0>; <0, 0, 4, 0>; <0, 0, 0, 4> and <2, 2, 2, 2>. The first four vectors in the list have $\Sigma X_i = 4$ and the last has $\Sigma X_i = 8$. We will now show that $P(\Sigma Z_i = 16 \mid \Sigma Y_i = 16 \ \& \ \Sigma X_i = 4)$ is not in general equal to $P(\Sigma Z_i = 16 \mid \Sigma Y_i = 16 \ \& \ \Sigma X_i = 8)$. For example, if $\mathbf{X} = \langle 4, 0, 0, 0\rangle$, then $\Sigma Z_i = 16$ if and only if $W_1 = 1$, where $P(W_1 = 1) = p$. Similarly for the vectors <0, 4, 0, 0>, <0, 0, 4, 0> and <0, 0, 0, 4>. Given that $\Sigma X_i = 4$ and $\Sigma Y_i = \Sigma((X_i)^2 = 16$, the set of the first four vectors has probability 1, and each individual vector of the first four has probability 0.25. Therefore $P(\Sigma Z_i = 16 \mid \Sigma Y_i = 16 \ \& \ \Sigma X_i = 4) = p$. On the other hand, the probability that $\mathbf{X} = \langle 2, 2, 2, 2\rangle$ is 1 given that $\Sigma X_i = 8$ and $\Sigma Y_i = \Sigma((X_i)^2 = 16$. Therefore $P(\Sigma Z_i = 16 \mid \Sigma Y_i = 16 \ \& \ \Sigma X_i = 8)$ is just the probability that $W_i = 1$ for $i = 1, 2, 3, 4$, which is p^4.

Much about the preceding example—e.g., that $n = 4$, that X is uniformly distributed, that X has 5 distinct values, that $Y = X^2$—is obviously inessential; $Y = X^2$ was used only because it is the simplest non-linear, non-Boolean function proposed for a regulator (Schilstra, 2002). Similar arguments would apply to a variety of non-linear dependencies of Y on X.

The considerations we have advanced in this section argue that, other than by chance, genetic regulatory network inference from associations among measured expression levels is possible only if conditional independence relations in the individual cells are (approximately) preserved in sums of those i.i.d. units. Although the particular example we gave was not biologically relevant, there are biologically relevant cases in which those conditional independence relations are not preserved. Chu (2004) has provided general sufficient conditions for conditional independence relations not to be invariant.

There are conditions under which the conditional independence relations among the summed expression levels of the genes from large number of cells will eventually be determined by the covariance matrix of the expression levels of the genes within a single cell. Recall that unlike conditional independence relations, the covariance matrix, with appropriate normalization, is invariant under aggregation. Those conditions are given in Theorem 3.

Theorem 3: Let $\{(X_n, Y_n, \mathbf{Z}_n)\}$ be a sequence of i.i.d. $k+2$ dimensional random vectors with mean $\mathbf{0}$ and nonsingular covariance matrix $\mathbf{\Sigma}$. Suppose $(X_n, Y_n, \mathbf{Z}_n)$ and $\mathbf{Z}_n$ both have bounded densities (with respect to the Lebesgue measure). Let $X_n^* = \Sigma X / \sqrt{n}$, $Y_n^* = \Sigma Y / \sqrt{n}$, and $\mathbf{Z}_n^* = \Sigma \mathbf{Z} / \sqrt{n}$, and $(U, V, \mathbf{W})$ be a

multivariate normal random vector with mean **0** and covariance matrix **Σ**. Then the total variation distance between (i) the conditional distribution of (X_n*, Y_n*) given $\mathbf{Z}_n$*, and (ii) the product of the conditional distributions of X_n* given $\mathbf{Z}_n$* and Y_n* given $\mathbf{Z}_n$* converges to: the total variation distance between (i) the conditional distribution of (*U*, *V*) given **W**, and (ii) the product of the conditional distributions of *U* given **W** and *V* given **W** almost surely with respect to the measure induced by **W**.

The implication of Theorem 3 is that, assuming we can model the gene expression levels as continuous random variables satisfying some regularity conditions, the conditional independence relations among the summed expression levels of a large number of cells are determined by the covariance matrix of the summed expression levels, regardless of the conditional independence relations among the gene expression levels in a single cell. For example, if in a single cell, gene *X* and gene *Y* are independent given gene *Z*, but the partial correlation between *X* and *Y* given *Z* is non-zero—this is usually the case when (*X*, *Y*, *Z*) do not follow a multivariate normal distribution—then given *n* such cells, Σ*X* and Σ*Y* are dependent given Σ*Z*. (Note that the correlation matrix of the gene expression levels is preserved under aggregation.)

While the conditions for Theorem 3 seem to be quite general, they do not cover the class of discrete distributions. After all, the expression level of any gene in a cell—the number of mRNA transcripts for that gene at a moment—is an integer-valued random variable. Continuous distributions can approximate a discrete distribution arbitrarily well, though only in terms of the distribution function. Theorem 3 can, however, be extended to an important class of discrete distributions—the regular lattice distributions—which covers the possible distributions of the numbers of mRNA transcripts of any set of genes in a cell.

A lattice distribution for a random vector **X** is a discrete distribution that only assigns non-zero probabilities to points $\mathbf{x} = (x_1, \ldots, x_k)$ such that $x_i = mh_i + b_i$, where *m* is an integer, h_i a positive real value, and b_i a constant. If h_i is the largest positive real number such that X_i can only take values of the form $mh_i + b_i$, h_i is called the span of X_i. A regular lattice distribution is defined as: Suppose a random vector $\mathbf{X} = (X_1, \ldots, X_k)$ has a lattice distribution, and h_i is the span of X_i. **X** has a *regular* lattice distribution if, for each *i*, there are at least two vectors $\mathbf{x}^i = (x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_k)$ and $\mathbf{y}^i = (x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_k)$, such that $|y_i - x_i| = h_i$, $P(\mathbf{X} = \mathbf{x}^i) > 0$, and $P(\mathbf{X} = \mathbf{y}^i) > 0$.

Chu (2004) extended Theorem 3 to cases in which $\mathbf{Z}_n$ has a regular lattice distribution, not a bounded density. These results provide a sufficient condition for the conditional independence relation between genes *X* and *Y* given genes **Z** in a single cell *not* to be invariant under aggregation: namely, when the partial correlation of *X* and *Y* given **Z** is non-zero. Notice, however, that the partial correlation is invariant under aggregation.

These theoretical results appear to conflict with many reports of successful machine learning searches for regulatory structure. In many cases, however, the successes are with simulated data in which the simulated values for individual cell representatives are not summed in forming the simulated measured values, and are therefore unfaithful to the actual measurement processes. In several other cases, results with real data are not independently confirmed, but merely judged plausible. Rarely, results are obtained that agree with independent biological knowledge; in these cases the actual regulatory structure among the genes considered may approximately satisfy invariance of conditional independence for summed variables, or the procedures may simply have been lucky. Feasible, economical techniques for measuring concentrations of transcripts in single cells could make machine learning techniques based on associations of expressions valuable in identifying regulatory structure. Techniques for the measurement of concentrations of mRNA species have recently become available (Elowitz, Levine,

Siggia, & Swain, 2002; Ginsberg, *et al*., 2004; Levsky, Shenoy, Pezo, & Singer, 2002; Rosenfeld, Young, Alon, Swain, & Elowitz, 2005), and seem the more appropriate venue for the application of machine learning methods

Experimental techniques that take advantage of immunoprecipitation, tagging of binding sites and regulatory proteins, binding site sequence homologies, and evolutionary preservation of regulatory mechanisms, are proving more fruitful. We may hope that machine learning techniques that are biased by such extra information may prove useful, and two recent examples in the literature suggest that this hope may bear fruit.

Pe'er, Tanay, & Regev (2006) focus on learning the structure of what they term *regulation graphs*: those in which (i) a small subset of vertices are regulators; (ii) a non-regulator is not the parent of any other vertex; and (iii) the number of parents of any vertex is bounded by some small number. These graphical structures correspond to those in which there are a few regulators that control the activity of all other genes, and perhaps influence each other. The restriction to this relatively small set of possible graphs results in provable performance guarantees for their MinReg algorithm, but it is computationally intractable unless the set of possible regulators is small. Microarray data alone do not suffice to determine a small set of possible regulators, and so Pe'er, *et al*. (2006) use additional sources of information—functional annotations from other experiments, and sequence homologies—to restrict the possible regulator set. When these additional pieces of information are used for real-world data, the MinReg algorithm recovers more information than other algorithms; insufficient information is provided to accurately judge its absolute performance. The simulation tests of the MinReg algorithm cannot be evaluated since they do not use aggregated data (see next section).

Hartemink (2006) similarly uses auxiliary data both to constrain the set of possible graphs and to provide a prior bias. Hartemink's algorithm is a standard Bayesian learning algorithm in which prior knowledge is incorporated through a bias in the prior probability over possible graphs. He focuses on two different settings: one is a (relatively) static system that can be modeled using an acyclic graph; the other is a dynamic system that is modeled using a dynamic Bayesian network (essentially, a graph in which the variables are time-indexed). Auxiliary information is used to restrict the possible graphs through both (i) variable selection, as the two systems have only 32 and 25 variables picked out by biological function; and (ii) graphical restrictions on the dynamic network, as the current time step is assumed to be directly influenced only by the previous time step. Given these sets of possible graphs, transcription factor binding location data are used to provide a significant bias on the prior probabilities over the graphs. In the actual applications to real-world data (judged by comparison to a gold-standard network), the location data play a major role: the algorithm performs quite poorly when expression data alone are used. In fact, close consideration of the real-world results suggests that the location data is doing almost all of the work. The algorithm's performance using both types of data is only marginally superior to using location data alone.

EXPERIMENTAL RESULTS

As elaborated in the previous section, microarray measurements are from aggregates of thousands of cells, and conditional independence relations that hold for biologically realistic probability distributions in individual units are typically not the same as those that hold in the probability distribution for cell aggregates. There are at least seven other challenges facing algorithms for automated learning of

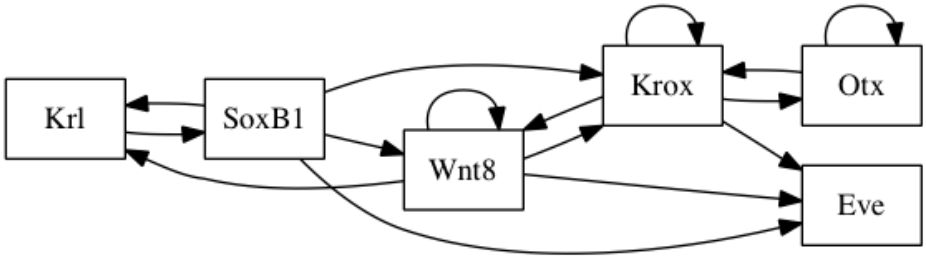
regulatory network structure: (1) the number of measurements of each gene is typically much smaller than the number of genes under study, and the number of genes—or genes at time points in time series representations—effectively defines the number of variables; (2) microarray measurements have a small signal to noise ratio; (3) many algorithms are based on acyclic graph representations which cannot faithfully represent both the probability distributions for equilibrium distributions of feedback systems and the mechanisms that lead to an equilibrium, and the only provably correct algorithm for learning arbitrary cyclic graphs from equilibrium data that is currently available (Richardson, 1996) has never been tested on gene expression data; (4) statistical associations among measured expression levels for different genes may depend on variations in unrecorded "common cause" regulator genes, or on extra-genetic factors not in the database; (5) summing variable values over many cell units reduces their variance, resulting in low correlations due to regulatory interaction, which implies the need for either very large samples or very large expression differences to reliably distinguish zero from non-zero correlations; (6) discretization of continuous variables can alter the original conditional independence relations among variables; and (7) when there are unrecorded sources of covariation, linear regression techniques overfit in the linear case, positing false connections even without correlated errors (Spirtes, *et al*., 2001), and feedback can produce statistical dependencies among measured variables similar to the effects of omitted common causes.

There are reports of successes at network inference with machine learning methods applied to both real-world and simulated expression data, but to our knowledge, no published simulation studies generate their data from experimentally established networks and treat measured values as aggregates of many individual cell values. In this section, we use realistic, aggregated, simulated data to examine the performance of nine structure learning algorithms: Reveal (Liang, *et al*., 1998), Bool2 (Akutsu, *et al*., 2000), MRBN (Friedman, Nachman, & Pe'er, 1999), PC (Spirtes, *et al*., 2001), CCD (Richardson, 1996), and algorithms described in Spirtes & Meek (1995), Arkin, Shen, & Ross (1997), D'Haeseleer, Wen, Fuhrman, & Somogyi (1999), Weaver, Workman, & Stormo (1999), and van Someren, Wesseksm & Reinders (2000). The version of MRBN we use was implemented by Aaron Darling (see http://mrbn.dyndns.org/), as other downloadable versions did not run, reimplementation was not possible from published accounts, and the authors did not respond to requests for clarification. PC and CCD were obtained from http:/www.phil.cmu.edu/projects/tetrad. The Meek/Spirtes algorithm was provided by Peter Spirtes from an old implementation not currently publicly available. We implemented the Reveal and Bool2 algorithms from published descriptions. The remaining algorithms were obtained from http://genlab.tudelft.nl/info.

These algorithms include procedures that discretize variables to binary or ternary values (Reveal, Bool2, MBRN), procedures that treat variables as continuous, procedures that use optimization routines (Bool2), regression procedures of various kinds (Weaver, van Someren, Arkin, D'Haeseleer), constraint based searches (PC, CCD), Bayesian scoring searches (MBRN) and hybrid constraint/Bayesian searches (Spirtes/Meek). Clearly these are not all of the algorithms that have been or could be proposed for studying gene regulation. For example, we have not applied the FCI algorithm (Spirtes, *et al*., 2001), nor have we included simulated annealing algorithms (Hartemink, 2001) or heuristic scoring procedures for Bayes nets with time indexed variables. We attempted to include a recent algorithm proposed and applied by Pe'er, Regev, & Tanay (2002), but they declined to provide their implementation.

This study used four datasets—three with simulated data, and one with experimental data:

Figure 2. The "maternal and early interactions" portion of the regulatory network of the sea urchin embryo as described in Davidson, et al. (2002)



1. Data generated in ten steps from a time series network modeling regulation in a fragment of the sea urchin genome (see Figure 2);
2. Data similar to (1) but projected to binary values;
3. Data similar to (1) but projected to three values;
4. Data from microarray measurements of variations of expression levels over the cell cycle in yeast (Spellman, *et al*., 1998) compared with a recent experimental determination of a substantial fraction of the regulatory network in the same species (Lee, *et al*., 2002).

The first three datasets are based on the multi-year effort by Davidson and his collaborators (Davidson, *et al*., 2002) to elucidate the genetic network of the sea urchin embryo, resulting in experimental data for a network of some forty genes. We developed a Java implementation of the "maternal and early interactions" portion of the sea urchin network, at least as it was understood at the time, using realistic transfer functions relating gene inputs to their outputs (see Figure 2; note that there are six genes: Wnt8, Krl, SoxB1, Krox, Otx, and Eve). This network has several feedback loops, including three genes that directly auto-regulate. We note in passing that the "truth" of the simulated network is irrelevant to the point at hand; all that matters is that this network involves realistic connectivity and realistic transfer functions. To simulate measurement noise, we multiplied the output value for each gene by the value of a random Gaussian variable with mean 1 and variance 0.01. We did not include additive error.

Our Java implementation realized a detailed reconstruction of the transfer functions and other features of the network as implemented in a NetBuilder model of the maternal and early interactions portion of the organism. NetBuilder is well documented (Schilstra, 2002), as is its model of this organism. It allows a user to "build" a gene network and to specify complex, non-linear and Boolean transfer functions. We built a NetBuilder version of the network under study and carried out a comparison of our code's calculations with NetBuilder's output over a number of steps; the results agreed closely. Christophe Battail has published a web page comparing NetBuilder's simulation of the Endo16 gene with experimental results (http://strc.herts.ac.uk/bio/maria/NetBuilder/Examples/Endo16/Endo16sim02.htm).

To create a *non-aggregated dataset*, we recorded the simulated expression level for each of the six genes (Wnt8, Krl, etc.) at each of up to 10 time steps; we call the values recorded for one such run a *non-aggregated sample*. The data matrix thus has simulations in rows, and each gene-time step as a column (variable). In the description of results below, *S* denotes the number of non-aggregated samples in a particular non-aggregated dataset. To construct an *aggregated sample*, we compute the mean for each column of a non-aggregated dataset; each aggregated sample thus corresponds to a full, non-aggregated

dataset. An *aggregated dataset* is a collection of such aggregated samples. In the results below, *R* denotes the number of non-aggregated datasets used to construct each aggregated sample in an aggregated dataset. In all of our experiments the sample sizes are comparatively small—reflecting the reality of microarray studies—and in most cases, the distributions are non-Gaussian, and the dependencies are non-linear. We approximated mean-zero normality by taking logs of all values in the data matrices (for both non-aggregated and aggregated datasets) and then subtracting the median of each column from all the values in that column.

By projecting based on the median value of each variable, we binarized the same data for tests of the Reveal and Bool2 algorithms. The MRBN algorithm implementation automatically projects real values to one of three values. The PC and CCD algorithms require multiple samples, each consisting of an entire time series. The binary algorithms require as input a dataset consisting of binary values for each of a set of genes at each of a number of time steps, and so the same time series can be used for comparisons. All datasets are publicly available at http://www.phil.cmu.edu/projects/genegroup.

The fourth dataset comes from four experiments (Spellman, *et al*., 1998) in which mRNA expression levels were measured in the course of the cell cycle with cells synchronized in different ways (see http://staffa.wi.mit.edu/cgi-bin/young_public/navframe.cgi?s=17&f=downloaddata). Friedman, *et al*. (2000) applied the MRBN algorithm to this data to obtain conjectured regulatory relations among the genes. Comparison experimental data are from Lee, *et al*. (2002), who applied immunoprecipitation techniques to experimentally estimate genes directly regulated by each of more than 100 known yeast regulators. The different Spellman experiments used samples in different metabolic conditions, so it is not sensible to use them as repeated samples of the same time series. We therefore simply concatenated the data so they appeared to be from one experiment; this introduces 3 false breaks in time series.

PC, CCD and Spirtes/Meek take a significance level as input; we give results for .05, but results for other significance levels up to 0.3 are similar, with lower significance levels slightly better in most experiments. Results for the van Someren and D'Haeseleer algorithms were essentially identical and we show only the latter.

To assess the performance of the algorithms, we ignored edge direction and focused on only the (simpler) problem of determining adjacency relations. An adjacency is judged present between two genes in an algorithm output if and only if it is present between those genes for any two times. Many other counting procedures are possible within each experiment (e.g., majority rule; restriction to sequential time steps) that would reduce false positives and increase false negatives. There are twenty-one possible pairs of adjacent variables (since a gene can auto-regulate), and twelve of those pairs are actually adjacent in the true graph for the maternal and early interactions portion of the sea urchin embryo network. Random assignment of edges for pairs of genes would result in 10.5 expected errors and an error rate of 0.5. Simply saying "yes" to each possible adjacency would result in an error rate of 0.43. Tables 1-5 report mean performance (and variance in parentheses) over 10 replications of the simulation settings; note that the variance for false positives determines the variance of correct negatives (and similarly for false negatives and correct positives).

The results for the Spellman, *et al*. (1998) data are reported in Table 6. These data were restricted to the 11 cell cycle genes that appear in the diagram published by Lee, *et al*. (2002). We applied the PC and CCD algorithms to the data as though it were equilibrium data, using the 11 genes as variables, implicitly violating i.i.d. sampling assumptions of these algorithms. There are 66 possible regulatory relationships, ignoring direction of regulation, including autoregulation. Random assignment would thus imply 33 errors (and an error rate of 0.5).

These simulations clearly demonstrate that no confirmation of an algorithm for obtaining regulatory structure from expression data can be rationally justified by results with simulated data unless the data generating model is non-linear, with feedback, and the variable values are aggregated over simulated individual cells. Selective comparisons with independent wet-laboratory results do not suffice either. Among the tests reported here, Tables 2 and 3 describe the most realistic simulations, and the best tests, of the algorithms considered. Even so, in several respects the inference problems posed for those simulated datasets are easier than with real data: the correct time sampling frequency is known; all replications are with the same simulated metabolism and the same time sampling; there are no missing values; and the variables are aggregated over only 20 units (the larger the number of units of aggregation, the smaller the correlation among the aggregated variables).

The implementations of Reveal and Bool2 limited them to three regulators per gene. For those yeast genes actually with three or fewer regulators in the Lee, *et al*. (2002) model, the results for these algorithms were almost always at chance, indicating the restriction to three regulators was inessential to their performance. One run of Bool2 that allowed for up to four regulators was attempted for the yeast data; the program ran for about 8 hours (over 200 times as long as the three regulator case) and returned the null model (no estimated regulatory relationships). On the simulation data, the Reveal, Bool2, Weaver, and D'Haeseleer algorithms proved useless; the remaining algorithms proved to be of some slight utility.

Table 1. Non-aggregated datasets; S = 20 (i.e., 20 non-aggregated samples in each dataset)

	False Pos	Correct Pos	False Neg	Correct Neg	Total Errors	Error rate
PC05	5.2 (1.3)	9.2	2.8 (0.6)	3.8	8.0 (2.2)	0.38
CCD05	4.4 (2.7)	7.6	4.4 (2.3)	4.6	8.8 (5.5)	0.42
Meek	4.2 (2.2)	10.4	1.6 (0.5)	4.8	5.8 (4.2)	0.28
Reveal	6.0 (0.1)	8.2	3.8 (0.2)	3.0	9.8 (0.4)	0.47
Bool2	6.1 (0.6)	7.9	4.1 (0.1)	2.9	10.2 (0.3)	0.49
MRBN	3.2 (1.0)	3.2	8.8 (2.6)	5.8	12 (3.7)	0.57
Arkin	1.4 (0.02)	2.7	9.3 (0.02)	7.6	10.7 (0.01)	0.51
Weaver	8.9 (0.01)	11.9	0.1 (0.0)	0.1	9.0 (0.01)	0.43
D'Haeseleer	5.4 (0.09)	7.3	4.7 (0.14)	3.6	10.1 (0.11)	0.48

Table 2. Aggregated samples; S = 20 and R = 30 (i.e., 30 aggregated samples)

	False Pos	Correct Pos	False Neg	Correct Neg	Total Errors	Error Rate
PC05	5.4 (1.8)	8.9	3.1 (2.1)	3.6	8.5 (2.5)	0.40
CCD05	4.9 (0.54)	8.5	3.5 (2.5)	4.1	8.4 (3.2)	0.40
Meek	5.7 (0.9)	9.1	2.9 (0.8)	3.3	8.6 (1.2)	0.41
Reveal	6.1 (0.0)	8.1	3.9 (0.07)	2.9	10 (0.17)	0.48
Bool2	6.3 (0.06)	7.8	4.2 (0.0)	2.7	10.5 (0.1)	0.50
MRBN	1.9 (1.0)	4.0	8.0 (0.4)	7.1	9.9 (2.1)	0.47
Arkin	1.4 (0.1)	2.7	9.3 (0.05)	7.6	10.7 (0.26)	0.51
Weaver	9.0 (0.0)	11.9	0.1 (0.0)	0.0	9.1 (0.0)	0.43
D'Haeseleer	5.3 (0.07)	7.0	5.0 (0.14	3.7	10.3 (.34)	0.49

Table 3. Aggregated samples; S = 20 and R = 100

	False Pos	Correct Pos	False Neg	Correct Neg	Total Errors	Error Rate
PC05	5.7 (1.8)	9.3	2.7 (1.8)	3.3	8.4 (3.6)	0.40
CCD05	5.4 (0.7)	8.8	3.2 (2.2)	3.6	8.6 (4.0)	0.41
Meek	5.6 (0.9)	9.1	2.9 (1.9)	3.4	8.5 (4.5)	0.40
Reveal	6.1 (0.0)	8.2	3.8 (0.0)	2.9	9.9 (0.0)	0.47
Bool2	6.2 (0.0)	7.8	4.2 (0.0)	2.8	10.4 (0.0)	0.50
MRBN	2.2 (1.3)	3.7	8.3 (1.1)	6.8	10.5 (1.4)	0.50
Arkin	1.5 (0.04)	2.7	9.3 (0.09)	7.5	10.8 (0.2)	0.51
Weaver	9.0 (0.0)	12.0	0.0 (0.0)	0.0	9.0 (0.0)	0.43
D'Haeseleer	5.5 (0.18)	7.3	4.7 (.17)	3.5	10.2 (0.16)	0.49

Considering both positive and negative errors, the Reveal, Bool2, MRBN and Arkin algorithms performed essentially at chance in all experiments: they are equivalent to flipping a coin to decide adjacencies. For non-linear simulated data, Weaver's algorithm is equivalent to saying "yes" to every adjacency; for linear data one would do better to use an inverted Weaver algorithm: say "yes" when it says "no." The D'Haeseleer algorithm is better than chance only for linear data, where it approximates saying "yes" in almost all cases. The PC and CCD are a little better than chance in all experiments and considerably better than chance with linear data. The Meek/Spirtes hybrid algorithm nearly dominates for total error rates on simulated data, and shows the theoretically expected increase in false positives with aggregated non-linear data. None of the algorithms improved with sample size increases up to 100. If we consider only the ratio of correctly predicted positives to predicted positives, and the most realistic simulations (Tables 2 and 3), the PC, CCD, Meek/Spirtes, MRBN and Arkin algorithms all do slightly better than merely saying "yes" in all cases, varying in Table 4 from .62 to .64 as against the constant "yes" ratio of .57. The MRBN and Arkin algorithms purchase the slight improvement at the cost of missing most of the true positives.

These results tend to confirm the theoretical arguments against the reliability of machine learning algorithms for estimating gene regulation networks from microarray measurements of expression lev-

Table 4. Non-aggregated samples; S = 100 and linear transfer functions were used

	False Pos	Correct Pos	False Neg	Correct Neg	Total errors	Error rate
PC05	1.2 (1.1)	6.9	5.1 (1.4)	7.8	6.3 (2.7)	0.30
CCD05	1.4 (0.7)	7.3	4.7 (0.7)	7.6	6.1 (2.3)	0.29
Meek	1.8 (1.5)	7.5	4.5 (0.95)	7.2	6.3 (3.1)	0.30
Reveal	5.2 (0.0)	6.8	5.2 (0.0)	3.8	10.4 (0.0)	0.50
Bool2	5.3 (0.0)	7.6	4.4 (0.0)	2.7	9.7 (0.0)	0.46
MRBN	1.9 (0.5)	3.4	8.6 (1.4)	7.1	10.5 (1.2)	0.50
Arkin	1.2 (0.0)	3.2	8.8 (0.03)	7.8	10.0 (0.07)	0.48
Weaver	4.5 (0.17)	3.8	8.2 (0.1)	4.5	12.7 (0.02)	0.60
D'Haeseleer	8.0 (0.03)	11.3	0.7 (0.02)	1.0	8.7 (0.03)	0.41

Table 5. Aggregated samples; S = 20 and R = 100; linear transfer functions were used

	False Pos	Correct Pos	False Neg	Correct Neg	Total errors	Error rate
PC05	1.2 (0.84)	7.4	4.6 (0.84)	7.8	5.8 (3.7)	0.28
CCD05	1.7 (0.9)	7.5	4.5 (1.6)	7.3	6.2 (3.0)	0.30
Meek	2.0 (0.9)	7.5	4.5 (0.7)	7.0	6.5 (1.6)	0.31
Reveal	5.3 (0.0)	6.9	5.1 (0.0)	3.7	10.4 (0.0)	0.50
Bool2	5.3 (0.0)	7.6	4.0 (0.0)	3.7	9.7 (0.0)	0.46
MRBN	2.1 (0.77)	3.0	9.0 (1.6)	6.9	11.1 (1.9)	0.53
Arkin	1.2 (0.07)	3.2	8.8 (0.03)	7.8	10.0 (0.07)	0.48
Weaver	5.6 (0.03)	4.7	7.3 (0.02)	3.4	12.9 (0.01)	0.61
D'Haeseleer	7.8 (0.03)	11.2	0.8 (0.03)	1.2	8.6 (0.08)	0.41

Table 6. S. Cerevisiae data from Spellman et al. (2002)

	False Pos	Correct Pos	False Neg	Correct Neg	Total errors	Error rate	
PC05	5	3	26	32	31	0.47	
CCD05	5	3	26	32	31	0.47	
Reveal	16	13	16	21	32	0.48	
Bool2	2	1	28	35	30	0.45	
MRBN	18	6	23	19	41	0.62	
Arkin	3	2	27	34	30	0.45	
Weaver	12	18	11	25	23	0.35	
D'Haeseler	2	1	28	35	30	0.45	

els. The Meek/Spirtes algorithm, which does notably better than chance or constant "yes" responses on non-linear, non-aggregated data in Table 1, falls to the constant "yes" error rate when the variables are aggregated. The linear regression procedures overfit with data from a linear feedback system. It is conceivable that other counting principles would decrease false positives for PC, CCD and Spirtes/ Meek and perhaps other algorithms in Tables 2 and 3, rendering them more useful, but we have not explored the possibilities. It would be preferable to have each algorithm run by its authors on common, well-specified, realistic simulation data from structures kept secret from those executing the algorithms and with explicit, pre-specified principles for counting errors, but such cooperative tests seem unlikely while relevant authors make neither their algorithms nor implementations publicly available.

COMPLEXITY WITHOUT CONDITIONAL INDEPENDENCE

The previous sections considered learning regulatory network structure from conditional independence information in microarray measurements. Network structure learning can instead be based on comparisons between the expression levels of various genes in (i) "wild type" cells that are not experimentally manipulated; and (ii) strains in which the expression levels of various genes have been suppressed or enhanced.

This strategy essentially follows the logic of standard causal inference from experimental interventions and controls, though supplemented with algorithms that attempt to extract maximal information from the data. If we manipulate gene G_1 and the expression level of G_2 changes, then G_1 must a cause—direct or indirect—of G_2. Moreover, this strategy does not use conditional independence information, and so the theoretical and experimental results about aggregation are irrelevant.

In principle, N experiments, each manipulating one of N genes, would suffice to identify the entire network if it is acyclic (without feedback), all effects are transitive, and no gene is both a direct and indirect regulator of any other gene. If $G(i)$ is the set of genes whose expression levels are altered when the expression level of gene i is experimentally randomized, then $G_k \rightarrow G_j$ if and only if $G(j) \subset G(k)$ and there is no r such that $G(j) \subset G(r) \subset G(k)$. If we can manipulate multiple genes at a time and trust conditional independence information, then the number of experiments can be reduced to around $\log_2(N)$ (Eberhardt, 2007), but for reasons noted above these methods are not applicable to real-world gene regulation data with, e.g., feedback (Frenster & Hovsepian, 2002).

This section considers the number of experiments required when cyclic network graphs are possible, and we set aside the statistical difficulties in determining differential expression (though we return to those issues at the end of this section). We also consider here the practical question of which experiment should be performed next, given the current state of one's knowledge. In contrast to suggestions in some of the literature (e.g., Ideker, Thorsson, & Karp, 2000; Onami, Kyoda, Morohashi, & Kitano, 2001), we argue that experimental manipulations do not permit efficient search for the true regulatory network; the computational complexity is too great. We continue to use graphical model representations of regulatory networks, but now allow cyclic graphs. We understand edges in terms of idealized experimental manipulations: $X \rightarrow Y$ in a network of genes $\mathbf{V}$ if and only if there are experimentally producible values of X, $x_1 \neq x_2$, such that the expression level of Y differs for x_1 and x_2 when all other genes are held fixed. We cannot in practice hold most gene expression levels fixed at some value, even the "wild type" level; we have only techniques to suppress or overexpress a gene. We therefore focus on experiments in which a subset of the genes are experimentally suppressed or overexpressed while one measures the expression levels of the other genes. These experimental limitations may prevent us from learning the precise structure of certain networks (e.g., if an indirect influence is only detectable by (impossibly) holding the mediating gene fixed at its wild type level).

This edge semantics implies an obvious inference principle: Infer $G \rightarrow H$ if and only if there are experiments E and E^* such that (i) G is manipulated to different values in E and E^*; (ii) H is not manipulated in E and E^*; (iii) H's expression level differs between E and E^*; and (iv) E and E^* do not differ in their treatment of any other variable. This inference principle is a precise statement of an obvious idea: gene i regulates gene j just when a change in the experimental manipulation of gene i (while not changing anything else in the system) leads to a change in the expression level of gene j. Crucially, this inference principle depends only on (significant) differences in expression levels, rather than on conditional independencies; as such, it is not subject to problems due to aggregation.

The worst-case complexity for number of experiments arises when none of the genes in the network regulate any others. An inference that G definitely does not regulate H requires finding that H has the same expression level for the three different experimental manipulations of G—wild type, suppressed, and overexpressed—for every combination of the three possible treatments for each of the other variables. Manipulation of other genes is required because of the possibility that G is a redundant regulator of H: some other gene L also regulates H, and so the influence of G is noticeable only for particular settings of L. G and L might alternately have a complex interaction in regulating H. Because of these

possibilities, we cannot simply manipulate G and look for a change in H. Exclusion of $G \rightarrow H$ from the network thus requires $3 \times 3^{n-2}$ distinct experiments. If no genes regulate any others, then one must conduct all of these experiments for every ordered pair of genes, and so $n \times (n - 1) \times 3 \times 3^{n-2}$ distinct experiments will be required in all.

Any reliable network inference algorithm must thus use information from exponentially many different experiments in the worst case, and so any reliable inference algorithm must itself have exponential complexity in the worst case. For $n = 9$ (as in the empirical data we consider below), reliable inference requires 472,392 experiments in the worst case. The worst-case number of experiments is required only if G does not regulate H; if G actually does regulate H, then that can be reliably discovered in as few as two experiments. The expected and real-world computational complexity of algorithms based on this inference principle will almost certainly be much less than the worst-case bound. In general, the algorithmic complexity decreases as the regulatory network density increases (i.e., as the number of edges goes up).

A different complexity analysis focuses on the number of networks that are consistent with some set of experiments, as well as the number of consistent, minimal networks. Suppose we have m distinct experimental conditions, each repeated l times, in which we measure the expression levels of n genes. Experimental conditions may differ in the genes that are suppressed or overexpressed, in various environmental conditions such as nutrient levels, or both. Let m_{ij} denote the mean expression level of gene i in experimental condition j, typically after normalizing distributions across conditions. We are most interested in a gene's expression level being different between two different experimental conditions, and so we assume that (using some simultaneous hypothesis test) we obtain a statistical decision about whether $m_{ij} = m_{ik}$ for each gene i and all pairs of experimental conditions j and k in which gene i is not directly manipulated. The j, k pairs for which $m_{ij} \neq m_{ik}$ are the findings that must be explained; the search problem is to find graphs that explain all of the observed expression level changes, and in the case of minimal networks, only the observed changes.

Formally, we construct a three-dimensional matrix $\boldsymbol{A}$ of size [gene & exogenous condition] × experimental condition × experimental condition. For each gene i:

- a_{ijk} = M, if either gene i is the target of an experimental manipulation in only one of j and k, or if it is manipulated in different ways in j and k;
- $a_{ijk} = 1$, if gene i is not manipulated in j and k, and the statistical decision is that $m_{ij} \neq m_{ik}$ (i.e., gene i has significantly different expression in the two conditions); and
- $a_{ijk} = 0$, otherwise (i.e., either gene i is experimentally manipulated in the same way in both conditions, or there is no significant difference in mean expression level).

For each exogenously controlled experimental condition h (e.g., temperature, nutrient level):

- $a_{hjk} = 1$, if h's value differs between conditions j and k; and
- $a_{hjk} = 0$, otherwise.

Define a graph G to be *consistent* with a set of experimental results $\boldsymbol{A}$ if and only if: (i) G does not contain any edges incompatible with $\boldsymbol{A}$ (in a sense defined below); and (ii) for all $a_{ijk} = 1$, there exists a gene q such that there is a directed path in G from q to i, and a_{qjk} = either 1 or M. The second condition ensures that G can explain every significant difference in expression levels. The first condition ensures

Table 7. Example expression data in three experiments

	Gene 1	Gene 2	Gene 3
Experiment 1	wt_1	wt_2	wt_3
Experiment 2	suppressed	wt_2	$wt_3 + \varepsilon$
Experiment 3	wt_1	suppressed	$wt_3 - \delta$

that *G* contains no impossible (relative to the data) edges. More precisely, we define a $G \rightarrow H$ edge to be *incompatible* with ***A*** if and only if: for all possible combinations of experimental manipulations (including no manipulation) of all genes except *G* and *H*, the expression level of *H* does not change regardless of the state of *G*. Note that an edge is only definitely incompatible with ***A*** if we perform every experiment in which *H* is unmanipulated (i.e., 3^{n-1} experiments); it is quite difficult to definitively rule out regulatory dependencies when we allow redundant pathways, cycles, and nonlinearities.

We are often most interested in the *minimally consistent* graphs for ***A***: those that are consistent with ***A***, but not consistent if any edge is removed. The *IG (Initial Graphs)* algorithm finds such graphs:

1) For each gene *i* and all *j*, *k* such that a_{ijk} =1, let L_{ijk} be the set of genes and exogenous factors *l* such that a_{ljk} = either 1 or M.
2) For each gene *i*, determine C_i: the set of minimal covering sets for all non-empty L_{ijk}.
3) Construct the collection of directed graphs, **G**, consisting of every possible graph that can be formed by choosing, for each *i*, some $c_i \in C_i$ and then making all factors in c_i into parents of *i*.
4) For each pair of experimental conditions *j*, *k*, let **I** be the set of factors such that a_{ijk} = M (or 1, in the case of exogenous factors). For each gene *r* such that a_{rjk} = 1, and for all $G \in \mathbf{G}$, if there is no directed path in *G* from a member of **I** to *r*, then replace *G* with all extensions of *G* that add a directed edge from a member of **I** to *r*.
5) Return **G** (henceforth, called **InitialGraphs**).

In plain language, steps 1 and 2 determine the "minimal" explanations for all differences in expression level. Step 3 then constructs the graphs corresponding to all possible combinations of minimal explanations. Finally, step 4 ensures that all of the explanations ultimately ground out in an experimental manipulation. Note that this algorithm can output cyclic graphs.

As an example, suppose Table 7 gives example expression data, where 'wt_i' indicates the wild type expression level of gene *i*. Suppose further that neither ε nor δ is itself a significant change (from wt_3), but $\varepsilon + \delta$ is a significant change in expression level. In that case, the corresponding ***A*** matrix is given in Table 8.

The expression levels of both genes 1 and 2 change significantly only between experimental conditions with different manipulations of that gene. Therefore, for those two genes, there are no sets to be

Table 8. Significant difference matrix for example experiments

a_{112} = M	a_{212} = 0	a_{312} = 0
a_{113} = 0	a_{213} = M	a_{313} = 0
a_{123} = M	a_{223} = M	a_{323} = 1

covered; C_1 and C_2 are both the empty set. For gene 3, there is only a significant change in expression level between experiments 2 and 3. L_{323} = {gene 1, gene 2} since both change, and so the minimal covering sets for gene 3 are {gene 1} and {gene 2}. The IG algorithm thus outputs two different graphs: 1 $\rightarrow$ 3 2; and 1 3 $\leftarrow$ 2.

We can prove the following theorem about the contents of IG algorithm's output:

Theorem 4 (IG consistency): Every graph in **InitialGraphs** is consistent with ***A***, and **InitialGraphs** contains all graphs that are minimally consistent with ***A***.

InitialGraphs typically does not include all of the consistent graphs. Provably, if G is consistent with ***A*** and $i \rightarrow j$ is not incompatible with ***A***, then the graph formed by adding $i \rightarrow j$ to G is consistent with ***A***. If our primary interest is in the *number* of consistent regulatory networks, then we can add together the number of supergraphs of each $G \in$ **InitialGraphs**, and then subtract out the graphs that are doubly-counted. This procedure scales up poorly, however, as it requires, for m initial graphs, calculating the size of 2^m–1 sets (i.e., supergraphs of all initial graphs, and all of the different overlaps). We can, however, use this strategy to compute a lower bound on the number of consistent graphs, since the number of supergraphs of a subset of **InitialGraphs** is necessarily less than or equal to the number of supergraphs of all of **InitialGraphs**. We can also determine the number of supergraphs of one graph in **InitialGraphs**, then the number of supergraphs of two graphs, then… until we include all of **InitialGraphs**. We can optimize the calculation by starting with the sparsest graph(s), since they will contribute the largest terms. Finally, since the computed lower bound increases (weakly) monotonically at each stage, we can run the procedure and stop at any time to get a lower bound.

The IG algorithm can also be used as the basis for a procedure to select which experiment to perform next. An algorithm to find the globally optimal next experiment would need to consider all possible sequences of unperformed experiments, the possible outcomes in each experiment in each sequence, and the number of consistent graphs for each possible outcome after each experiment in a possible sequence. This computation is hopelessly intractable; if we have conducted L experiments, then we must compute the exact number of consistent graphs at every branch-point and leaf in an exponentially branching tree of depth $(3^n - L)$ corresponding to a sequence of experiments, where there are $(3^n - L)!$ many sequences/trees. Alternatively, we could use the IG algorithm in a heuristic procedure to select the best next experiment. For some set **E** of possible experiments, compute for each experiment E the expected number of consistent graphs over all possible outcomes of E, and then perform the experiment E^* that minimizes this number. One can clearly add a probability distribution over outcome likelihood for an experiment, if that prior knowledge is available. This procedure appears more promising, but still faces significant challenges. It is a greedy search procedure, and so can lead to sub-optimal sequences of experiments. It is also still quite computationally complex: there will typically be exponentially many unperformed experiments, and we must compute the exact number of consistent networks (i.e., conduct an exponential calculation) for every possible outcome of every $E \in$ **E**.

We have given theoretical reasons to question whether the basic inference principle underlying manipulation experiments can be generalized to a reliable search procedure. These concerns might fail to be an issue for realistic networks, however, and so we now consider a real-world case. Ideker, *et al*. (2001) conducted a series of experiments on the galactose metabolism cycle in yeast (*Saccharomyces cerevisiae*). They used microarrays to measure expression levels of 5000 different genes, but focused on nine genes that had previously been identified as important in this cycle. They performed ten experiments (wild type measurements, plus single knockouts of each gene) in two environments (presence and absence of galactose). We focus here on the galactose present case.

Table 9. Minimal covering sets for galactose genes in Ideker, et al. (2001) data

gal1	*gal2*	*gal3*	*gal4*	*gal5*	*gal6*	*gal7*	*gal10*	*gal80*
10	7	7	1	1,3	1,2	10	1,6	4,5
2,6,7	10	10	3	1,4	2,7	1,3	1,2,80	4,6
2,7,80		1,2	5	3,6	2,10	1,2	1,2,5	4,10
		1,6	7	2,4,6	5,7	1,6	2,3,6	5,7
		1,80	10	4,6,7	5,10	1,80	2,7,80	6,7
		4,6		4,6,10	1,3,5		2,6,7	7,10
		2,5,6			1,5,80		4,6,7	
		5,6,80			2,3,5		2,3,5,80	
					2,5,80			

Ideker, *et al*. (2001) performed four replications of each experimental condition, with appropriate instrumental counter-balancing across replications, in order to improve the power of statistical tests. There were nonetheless many statistical challenges to constructing the ***A*** matrix that encodes significant differences in expression level between experimental conditions. The most significant issues were correlated errors in the measurements, and the large number of simultaneous statistical tests required for this 9-gene/10-experiment setup. For reasons of space, we do not go into details here; a full statement of the statistical procedures that we used can be found in Danks, Glymour, & Spirtes (2003).

Despite these challenges, the Ideker, *et al*. (2001) data are close-to-ideal for the IG algorithm. We have expression levels for all genes in all ten experiments, and so can carry out all of the required pairwise comparisons. Automated search is also clearly required for these data, since there are $2^{72} \approx 4 \times 10^{21}$ possible regulatory networks over these nine genes. The output of the first three steps of the IG algorithm (i.e., the minimal covering sets for each gene) are shown in Table 9. Each column gives the minimal covering sets for that particular gene, where each row within a column is a different minimal covering set.

This table describes 3,110,400 different cyclic and acyclic graphs corresponding to all possible ways of choosing parents (i.e., minimal covers) for each gene. After step (4) of the IG algorithm—checking whether every expression level change is explained in every graph—the IG procedure returns a set of 3,480,675 graphs.

The output of the IG algorithm includes all networks that are minimally consistent with ***A***, but not necessarily all consistent graphs. The sparsest graph in **InitialGraphs** contains fourteen edges, and no edges are inconsistent with the data. A lower bound (on the total number of consistent graphs) calculated using only this sparsest graph is approximately 2×10^{17} networks. The actual number of consistent graphs is almost certainly larger. These ten experiments have thus reduced the possibility space by (at most) four orders of magnitude, and probably much less.

This section began with the hope that differences in expression levels between experimental conditions could provide the basis for reliable search procedures. The theoretical and empirical results argue, however, that such data actually provide relatively little information: there are typically many different potential regulators for some expression level difference, and too many experiments are required to rule out a regulatory connection. At the same time, gene manipulation experiments can be expected to remain an important confirmatory method for testing a specific hypothesis. They are ineffective for search, but quite powerful for targeted hypothesis testing.

CONCLUSION

Machine learning methods offered the promise of a shortcut to discovering genetic regulatory networks. The promise has so far proved false, for reasons we have described. Machine learning methods—in particular, automated search for graphical causal models—are applicable to many other genomics problems, and potentially even to gene regulation problems when data are available at the individual cellular level. When our data come from aggregations of cells, however, then sophisticated machine learning methods are actually penalized, since the conditional independence relations on which they depend do not hold at the aggregate level. Simpler methods that do not exploit conditional independence relations (such as the IG algorithm in the previous section) are computationally intractable for realistic scenarios. There are of course multiple cases that purport to show successful machine learning from microarray data; these instances almost all (i) use unrealistic single-cell (simulated) data; (ii) focus on cases in which the ground truth is not known and so performance cannot be evaluated; or (iii) report only isolated successfully discovered regulatory connections, rather than statistics about overall algorithm performance. We thus conclude on a pessimistic note: despite the hopes of many (including us at a prior time), standard structure learning algorithms cannot be fruitfully applied to microarray data; rather, successful machine learning for genetic regulatory networks will depend on statistical, algorithmic, and experimental advances that are highly tuned to the challenges of this particular domain, and that largely remain to be done.

REFERENCES

Akutsu, T., Miyano, S., & Kuhara, S. (2000). Algorithms for inferring qualitative models of biological networks. *Pacific Symposium on Biocomputing, 5*, 290-301.

Arkin, A., Shen, P., & Ross, J. (1997). A test case of correlation metric construction of a reaction pathway from measurements. *Science*, *277*, 1275–1279. doi:10.1126/science.277.5330.1275

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B. Methodological*, *57*, 289–300.

Brown, C. T., Rust, A. G., Clarke, P. J. C., Pan, Z., Schilstra, M. J., & Buysscher, T. D. (2002). New computational approaches for analysis of cis-regulatory networks. *Developmental Biology*, *246*, 86–102. doi:10.1006/dbio.2002.0619

Chu, T. (2003). Learning from SAGE data. Unpublished doctoral dissertation, Carnegie Mellon University.

Chu, T. (2004). Limitations of statistical learning from gene expression data. *Interface 2004: Computational Biology and Bioinformatics*.

Chu, T., Glymour, C., Scheines, R., & Spirtes, P. (2003). A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays. *Bioinformatics (Oxford, England)*, *19*, 1147–1152. doi:10.1093/bioinformatics/btg011

D'haeseleer. P., Wen, X., Fuhrman, S., & Somogyi, R. (1999). Linear modeling of mRNA expression levels during CNS development and injury. *Pacific Symposium on Biocomputing, 4*, 41-52.

D'haeseleer. P. (2000) Reconstructing gene networks from large scale gene expression data. Unpublished doctoral dissertation, University of New Mexico.

D'haeseleer, P., Liang, S., & Somogyi, R. (2000). Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics (Oxford, England), 16*, 707–726. doi:10.1093/bioinformatics/16.8.707

Danks, D., & Glymour, C. (2002). Linearity properties of Bayes nets with binary variables. In J. Breese & D. Koller (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 17th conference (UAI-2001)* (pp. 98-104). San Francisco: Morgan Kaufmann.

Danks, D., Glymour, C., & Spirtes, P. (2003). The computational and experimental complexity of gene perturbations for regulatory network search. In W. H. Hsu, R. Joehanes & C. D. Page (Eds.), *Proceedings of IJCAI-2003 workshop on learning graphical models for computational genomics* (pp. 22-31).

Davidson, E., Rast, J., Oliveri, P., Ransick, A., Calestani, C., & Yuh, C. (2002). A genomic regulatory network for development. *Science, 295*, 1669–1678. doi:10.1126/science.1069883

Eberhardt, F. (2007). Causation and intervention. Unpublished doctoral dissertation, Carnegie Mellon University.

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science, 297*, 1183–1186. doi:10.1126/science.1070919

Frenster, J. H., & Hovsepian, J. A. (2002). RNA feedback mechanisms during eukaryotic gene regulation. In *Northwest symposium on systems biology* (p. 15).

Friedman, N., Nachman, I., & Pe'er, D. (1999). Learning Bayesian network structure from massive datasets: The 'sparse candidate' algorithm. In K. Laskey & H. Prade (Eds.), *Proceedings of the 15th international conference on uncertainty in artificial intelligence* (pp. 206-215). San Francisco, CA: Morgan Kaufmann.

Friedman, N., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Recomb 2000*, Tokyo.

Genovese, C., & Wasserman, L. (2001). False discovery rates (Tech. Rep. 762). Carnegie Mellon University: Department of Statistics.

Ginsberg, S. D., Elarova, I., Ruben, M., Tan, F., Counts, S. E., & Eberwine, J. H. (2004). Single-cell gene expression analysis: Implications for neurodegenerative and neuropsychiatric disorders. *Neurochemical Research, 29*, 1053–1064. doi:10.1023/B:NERE.0000023593.77052.f7

Hartemink, A. (2001). Principled search for gene regulation. Unpublished doctoral dissertation, Harvard University.

Hartemink, A. (2006). Bayesian networks and informative priors: Transcriptional regulatory network models. In K.-A. Do, P. Müller & M. Vannucci (Eds.), *Bayesian inference for gene expression and proteomics* (pp. 401-424). Cambridge: Cambridge University Press.

Hashimoto, R. F., Kim, S., Shmulevich, I., Zhang, W., Bittner, M. L., & Dougherty, E. R. (2004). Growing genetic regulatory networks from seed genes. *Bioinformatics (Oxford, England)*, *20*, 1241–1247. doi:10.1093/bioinformatics/bth074

Ideker, T., Thorsson, V., Ranish, J., Christmas, R., Buhler, J., & Eng, J. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, *292*, 929–934. doi:10.1126/science.292.5518.929

Ideker, T. E., Thorsson, V., & Karp, R. M. (2000). Discovery of regulatory interactions through perturbation: Inference and experimental design. *Pacific Symposium on Biocomputing*.

Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., & Gerber, G. (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, *298*, 799–804. doi:10.1126/science.1075090

Levsky, J. M., Shenoy, S. M., Pezo, R. C., & Singer, R. H. (2002). Single-cell gene expression profiling. *Science*, *297*, 836–840. doi:10.1126/science.1072241

Liang, S., Fuhrman, S., & Somogyi, R. (1998). Reveal: A general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, *3*, 18-29.

Onami, S., Kyoda, K. M., Morohashi, M., & Kitano, H. (2001). The DBRF method for inferring a gene network from large-scale steady-state gene expression data. In H. Kitano (Ed.), *Foundations of systems biology* (pp. 59-75). Cambridge, MA.: The MIT Press.

Pe'er, D., & Hartemink, A. (2004). Single-cell gene expression analysis: Implications for neurodegenerative and neuropsychiatric disorders. *Neurochemical Research*, *29*, 1053–1064. doi:10.1023/B:NERE.0000023593.77052.f7

Pe'er, D., Regev, A., & Tanay, A. (2002). MinReg: Inferring an active regulator set. In *Proceedings of the tenth international conference on intelligent systems for molecular biology (ISMB)*.

Pe'er, D., Tanay, A., & Regev, A. (2006). MinReg: A scalable algorithm for learning parsimonious regulatory networks in yeast and mammals. *Journal of Machine Learning Research*, *7*, 167–189.

Richardson, T. (1996). A discovery algorithm for directed cyclic graphs. In *Proceedings of the 12th international conference on uncertainty in artificial intelligence* (pp. 454-461).

Rosenfeld, N., Young, J. W., Alon, U., Swain, P. S., & Elowitz, M. B. (2005). Gene regulation at the single-cell level. *Science*, *307*, 1962–1965. doi:10.1126/science.1106914

Schilstra, M. (2002). NetBuilder software. Retrieved from http://strc.herts.ac.uk/bio/maria/

Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics (Oxford, England)*, *18*, 261–274. doi:10.1093/bioinformatics/18.2.261

Shrager, J., Langley, P., & Pohorille, A. (2002). Guiding revision of regulatory models with expression data. *Pacific Symposium on Biocomputing, 7,* 486–497.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., & Eisen, M. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell, 9,* 3273–3297.

Spirtes, P., Glymour, C., & Scheines, R. (2001). *Causation, Prediction and Search*. 2nd edition, Cambridge, MA: The MIT Press.

Spirtes, P., & Meek, C. (1995). Learning Bayesian networks with discrete variables from data. In U. M. Fayyad & R. Uthurusamy (Eds.), *Proceedings of the first international conference on knowledge discovery and data mining* (pp. 294-299). San Jose, CA: AAAI Press.

van Someren, E. P., Wessels, L. F. A., & Reinders, M. J. T. (2000). Linear modeling of genetic networks from experimental data. In *Proceedings of the eighth international conference on intelligent systems for molecular biology* (pp. 355-366).

Weaver, D. C., Workman, C. T., & Stormo, G. D. (1999). Modeling regulatory networks with weight matrices. *Pacific Symposium on Biocomputing, 4,* 112-123.

Yoo, C., Thorsson, V., & Cooper, G. F. (2002). Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. *Pacific Symposium on Biocomputing, 7,* 498–509.

Yuh, C., Bolouri, H., & Davidson, E. (1998). Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene. *Science, 279,* 1896–1902. doi:10.1126/science.279.5358.1896

Section 5

Analysis & Complexity

Chapter 14
Complexity of the BN and the PBN Models of GRNs and Mappings for Complexity Reduction

Ivan V. Ivanov
Texas A&M University, USA

ABSTRACT

Constructing computational models of genomic regulation faces several major challenges. While the advances in technology can help in obtaining more and better quality gene expression data, the complexity of the models that can be inferred from data is often high. This high complexity impedes the practical applications of such models, especially when one is interested in developing intervention strategies for disease control, for example, preventing tumor cells from entering a proliferative state. Thus, estimating the complexity of a model and designing strategies for complexity reduction become crucial in problems such as model selection, construction of tractable sub-network models, and control of the dynamical behavior of the model. In this chapter we discuss these issues in the setting of Boolean networks and probabilistic Boolean networks – two important classes of network models for genomic regulatory networks.

INTRODUCTION

One can think of a *Gene Regulatory Network (GRN)* as a network of relations among strands of DNA (genes) and the regulatory activities associated with those genes (Dougherty and Braga-Neto, 2006). This general definition allows for many mathematical (usually dynamical) systems to be called GRNs. The goodness of each such model is evaluated using several important criteria: the level of description of the biochemical reactions involved, complexity of the model, model parameter estimation, and the predictive power of the model. There have been many attempts to model the structure and dynamical behavior of GRNs, ranging from deterministic with discrete time space to fully stochastic with continuous time

DOI: 10.4018/978-1-60566-685-3.ch014

space. One can find a good review of such attempts in (de Jong, 2002). The so called central 'dogma' of molecular biology (Crick, 1970) implies that genes communicate via the proteins they encode. Both stages of protein production, transcription and translation, are controlled by a multitude of biochemical reactions, and are influenced by both internal and external to the cell factors. This perspective suggests that the expression of a given gene *i*, i.e. the quantity of either protein or messenger RNA, should be considered as a random function $X_i(t)$ of the cell's internal and external environments. Thus, if one wants to study the dynamical behavior of a GRN, one must design a mathematical model for the gene-expression vector $\boldsymbol{X}(t) = (X_1(t), X_2(t), \ldots, X_n(t))$ for the n genes that form the network. The stochastic differential equation model appears to provide the most detailed description of the dynamics of $\boldsymbol{X}(t)$. In principle, it could include all of the information about the biochemical processes involved in gene regulation. At the same time, the estimation of its parameters cannot be done without large amount of reliable time-series data. Thus, one is forced to take a more pragmatic approach and look for simpler models for the dynamics of the gene-expression vector. One of the most extreme simplifications is the *Boolean network* (BN) model, originally proposed by Kauffman (1969a). The BN model is based on the observation that during the regulation of its functional states the cell often exhibits switch-like behavior. Recent work using the NCI 60 Anti-Cancer Drug Screen has demonstrated that Boolean logic type interactions can be detected in gene expression data (Pal at al., 2005). While there are instances in gene regulation where the Boolean logic is the appropriate level of description of the interactions – for instance, when transcription factors have to form a complex that binds to the *cis*-regulatory DNA to activate transcription, one should keep in mind that discrete models cannot capture the details of the biochemical reactions involved in those processes. It is not the binary nature of the BN model that is its greatest weakness, one even more important deficiency is its determinism. Deterministic models, such as the BN, cannot represent the consequential perturbations due to external latent variables. In addition, the BN model cannot be used to represent biologically meaningful events, such as gene mutations. The stochastic extension of the BN model - *probabilistic Boolean network* (PBN), was introduced by Shmulevich at al. (2002b) in an attempt to account for those latent variables and gene perturbations while keeping the Boolean logic as the model for the gene-gene interactions. As a collection of BNs with a probability structure, the PBN model could be viewed as a minimal extension of the BN which allows for modeling of the stochastic nature of complex systems with lots of latent variables and random experimental effects. However, even such a minimal extension of the deterministic model exhibits high complexity which impedes its practical applications to model GRN of more than 20 genes. Hence, there is a need for constructing size reducing mappings that produce new and more tractable models that share some of the biologically meaningful properties of the larger-scale models. In addition, one needs to develop methods for complexity estimation of both the model and the mapping used to reduce its size.

In this chapter we use the BN and PBN models to illustrate how one can approach both problems: the complexity estimation of a model and the construction of size-reducing mappings for it. The term *complexity* is overloaded with many different meanings depending on the field of study. GRNs are composed of many parts that interact with each other and those interactions involve feedback loops and are stochastic from the observer's point of view. Thus, GRN satisfy the definition of a complex system, and one of the aspects of complexity-reduction strategies in this setting is to find subsystems with less number of interacting genes and with simpler rules of interactions between them which preserve, to a degree, important properties of the whole system. This kind of complexity is the main focus of the discussion in the chapter. A different approach to the complexity problem is based on the observation that every GRN can be viewed as an information processing machine which opens the door for applications

of the algorithmic information theory to study the complexity of a model for the GRN, and to compare the complexity of different models of the same underlying GRN. To our knowledge, there have been very few developments in this important direction of research. While the problem of estimating the algorithmic complexity of a given complexity-reduction mapping is important, we do not discuss it here because of space limitations, and refer the reader to the works listed as references. The discussion in the chapter will focus primarily on the design of reduction strategies producing more tractable models that preserve some of the important and biologically meaningful properties of the larger and more complex models. It is important to emphasize that the BN and the PBN models are used to set up and describe the so called *reduction problem* which can be similarly formulated in the case of different models of GRN. Our goal is not to compare different models of genomic regulation, nor to study the reduction problem in all possible different settings. We focus on presenting a framework that can be used to study mappings for complexity reduction and to point out some of the reasons why reducing the complexity of a GRN model is a hard problem, e.g. an ill-posed inverse one. There are several reasons for selecting BN and PBN models as examples of how complexity issues could be addressed. First, they are important discrete models that have been widely used in situations where groups of genes exhibit a switch-like behavior (Dougherty at al., 2005). Second, they are closely related to each other and to other models of GRNs, i.e. dynamic Bayesian networks (Lähdesmäki at al., 2006). Third, it has been suggested that some of the structural and dynamical properties of BNs and PBNs correspond closely to important biological characteristics of the living cells (Huang, 1999), (Kauffman, 1969a, 1993). Finally, these models are simple enough and yet exhibit rich dynamical behavior which is in concordance with Occam's razor principle given the paucity of data one usually deals with. The second major issue in this chapter is the complexity estimation of the model and the cost of applying size-reducing mappings to it. We discuss these problems using the notions of *stochastic complexity* and *Minimum Description Length* (MDL) principle (Rissanen, 2007).

BACKGROUND

The BN model was originally proposed by Kauffman (1969a, 1969b) as a framework for studying GRNs. It had been successfully used in physics before attracting the attention of the biology community. The initial application of the model was to study the evolution of ensembles of networks which were restricted to a specific type of fitness landscape. One can read more details about the BN model and its early applications in (Kauffman, 1973, 1993), (Huang, 1999), (Somogyi & Sniegoski, 1996), and (Glass & Kauffman, 1973). Here we provide the definition of a Boolean network and briefly discuss the ensemble approach.

Definition 1: A BN $B = (V, \boldsymbol{f})$ on n genes is defined by a set of nodes/genes $V = \{x_1,..., x_n\}$ and a vector of Boolean functions $\boldsymbol{f} = [f_1,..., f_n]$.

The variable $x_i \in \{0, 1\}$ represents the expression level of gene i, with 1 representing high and 0 representing low expression. The vector $\boldsymbol{f}$ represents the regulatory rules between genes. At every time step t +1, the value of x_i is predicted by the values of a set W_i of genes at the previous time step t, based on the regulatory function f_i, i.e. $x_i(t+1) = f_i\left(x_{i_1}(t),\ldots,x_{i_k}(t)\right)$. The set of genes $W_i = \{x_{i_1},\ldots,x_{i_k}\}$ is called the predictor set of x_i, and the function f_i is called the predictor function of x_i. The pairs (x_i, W_i), i = 1, ..., n induce a digraph G with edges $x_{i_j} \rightarrow x_i$ representing the structural dependencies among the

genes. A state of B is a vector $\boldsymbol{s} = [x_1,..., x_n] \in \{0,1\}^n$. All of the possible states of the Boolean network comprise its state space S which combined with the functions in $\boldsymbol{f}$ produces a digraph Γ called the state transition diagram of B. Γ represents the dynamics of the Boolean network and can be identified with a $2^n \times 2^n$ matrix P_n with rows and columns indexed by the states in B and entries $p_{ij} = 1$ or 0 if there is a transition from the state $s_i \rightarrow s_j$ in S. Given an initial state, the network will eventually enter a set of states in G through which it will repeatedly cycle forever. Each such set is called an attractor cycle, and a singleton attractor is an attractor cycle of length 1. The network attractors induce a partition of state space S where the subsets of states that belong to the same equivalence class is called the *basin* of the corresponding attractor cycle. The attractors of a Boolean network represent a type of memory of the dynamical system (Shmulevich at al., 2002b). In addition, one can interpret their dynamics as an abstract model of computation, which generalizes the processing of information done by cellular automata – a special case of Boolean networks (Codd, 1968), (Mitchell at al., 1994).

Originally, the BN model was used in biology and physics to study ensembles of randomly generated Boolean nets (Kauffman, 1993). Analytical results and numerical simulations focused on the relationships between the structural gene interdependencies and dynamical behavior of such ensembles have provided insights into the general characteristics of large GRNs and the related evolutionary principles. 'Tuning up' of ensemble parameters such as the *average connectivity K* and the predictor functions' *bias p* can be used to study the operating regimes of the networks. The average connectivity is defined as the average size of the predictor sets W_i, $K = \frac{1}{n}\sum_{i=1}^{n} k_i$ and the bias p is defined as the probability of a given predictor function to assume a value of 1. Depending on the values of K and p there are two main modes of operation of a BN: ordered and chaotic. In the ordered regime most of the system components/nodes are frozen at either 1 or 0 value, and the transfer of information is impeded by those large frozen islands of genes. In the chaotic regime, the system is very sensitive to small perturbations where a change of the value of one node can propagate to many others in an avalanche-like manner. The phase transition boundary between the ordered and the chaotic regimes is called the complex regime or critical phase. It has been shown that BNs in that regime are the most evolvable and Kauffman (1993) argues that life must exist on that edge between order and chaos: "a living system must first strike an internal compromise between malleability and stability. To survive in a variable environment, it must be stable to be sure, but not so stable that it remains forever static". *Structural stability* is one of the central concepts in the theory of dynamical systems. It describes persistent behavior that cannot be destroyed by small changes to the system. As real GRNs are capable of maintaining metabolic homeostasis and stable developmental program in the face of a changing environment, they certainly possess structural stability. The BN model naturally captures this phenomenon because the network 'flows' back to one of its attractors after a small gene perturbation. Following this line of reasoning, Kauffman (1993) suggests that the attractors in a BN correspond to cellular types. Another interpretation of the attractors of a BN (Huang, 1999, 2001) is that they represent cellular states, such as *proliferation* (cell cycle), *apoptosis* (programmed cell death), and *differentiation* (execution of cell-specific tasks). For example, if a structural perturbation (mutation) happens which moves the network from the basin of the apoptotic attractor, the cells could exhibit uncontrolled growth or hyper proliferation, typical of tumorigenisis. The two interpretations of the attractors in the BN model are complimentary to each other: for a given cell type, different functional states exist and are determined by the collective gene activity. Thus, a particular cell type can encompass several attractor cycles each one corresponding to different cellular

functional states. We refer the reader to (Kauffman, 1993), (Shmulevich at al., 2002b) and (Shmulevich at al., 2003b) for a detailed treatment and additional references to results about the interplay between the average connectivity and the bias of the predictor functions in a BN and how that impacts the dynamical behavior of the network. An important implication from the body of work on the effects of these local parameters on the network is that if one wants to model GRNs with BNs or their generalizations one should constraint the network connectivity in order to keep the model on the edge of chaos and closer to the ordered regime. For example, in the case of unbiased, $p = 0.5$, predictor functions the networks with $K > 2$ operate mostly in the chaotic regime which renders such models incompatible with the real GRNs which are clearly non-chaotic systems.

Although the ensemble studies can provide important insights into some general properties of the BN models, a single Boolean network itself is not capable of capturing the effects of latent variables or random gene perturbations. Moreover, the ensemble approach does not provide a way of explicitly inferring the specific BN structure from data, e.g. cDNA microarray gene expression. Inferring the BN structure from data has the potential to reveal how to design therapeutic intervention for GRNs which show a specific disease phenotype. The data used for network inference exhibits uncertainty on various levels. First, due to biological variability, gene expression is inherently stochastic. Second, the complex measurement process, the microarray preparation, image acquisition and processing create experimental noise that has to be taken into account during the inference of the network. All of this combined with the presence of latent or unobservable variables such as proteins or environmental conditions present us with the problem to infer deterministic predictor functions under uncertainty. To solve such a problem one needs to reliably estimate the uncertainty. Without such estimation one cannot be sure how the designed predictor function will perform when presented with new data. A possible approach is to follow the Occam's razor principle and to penalize the predictors that are too complex. Tabus at al. (2001, 2002) took such an approach using the well-known MDL principle and the *normalized maximum likelihood* (NML). A different approach was proposed by Shmulevich at al. (2002a). Keeping in mind that the predictor functions cannot be reliably estimated from the limited amount of data relative to the number of genes on a microarray slide, one can infer a number of simple predictor functions, each of which performs relatively well in predicting the target gene. Here, simpler is understood as having predictor sets W_i of smaller size. After producing such predictor functions, one has to combine them together accounting for the uncertainty at the same time. This 'probabilistic' approach to synthesize 'good' predictor functions leads to the PBN model of GRNs.

Definition 2: A *context-sensitive* PBN $A = A^{q,p}(V,\mathbf{F}, C)$ is defined by a set of nodes/genes $V = \{x_1,..., x_n\}$, a set of vector-valued Boolean functions $\mathbf{F} = \{\mathbf{f}_1,\ldots, \mathbf{f}_r\}$, $\mathbf{f}_j : \{0,1\}^n \rightarrow \{0,1\}^n$, $j = 1,\ldots, r$ called *realizations* or *network functions*, a list of selection probabilities $C = \{c_1,\ldots, c_r\}$ for the corresponding realizations, the gene mutation/flipping probability p, and the realization switching probability q.

Updating the values of all genes in the network at time t is done synchronously according to the components of the currently used network function, and then the process is repeated. The choice of which network function $\mathbf{f}_j$ to apply is governed by a selection procedure. Specifically, at each time point t a random decision is made as to whether to switch the network function for the next transition, with a probability q of a switch being a system parameter. If a decision is made to switch the network function, then a new realization is chosen from among all of the possible realizations $\mathbf{f}_j \in \mathbf{F}$ of A, according to their individual selection probabilities $c_j \in C$. In other words, each network function $\mathbf{f}_j$ represents a deterministic BN B_j and the PBN behaves as a fixed BN until a random decision (with probability of q)

is made to change the network function according to the probabilities $\{c_1,\ldots, c_r\}$ from among $\{\boldsymbol{f}_1,\ldots, \boldsymbol{f}_r\}$. In addition to the network switching and selection in the PBN model, there is mechanism which models random gene mutations, i.e. at each time point t there is a probability p of any gene changing its value uniformly randomly. Thus, the PBN model can account for the uncertainties in both data and model selection. The PBN A shares the same state space S with its realizations, and the state transition diagrams Γ_j of the individual B_j's combine naturally into a stochastic state transition diagram Γ representing the dynamics of A. As in the case of deterministic BNs Γ can be identified with a stochastic $2^n\times 2^n$ matrix P_n, also known as *transition* matrix, with non-negative entries p_{ij} and having the property $\sum_{j=1}^{n} p_{ij} = 1$, $i = 1,\ldots, n$. Using this matrix, the dynamics of A can be described using the well-developed theory of Markov chains. One should notice that if the probability of gene flipping p is positive then the Markov chain representing the dynamics of the network is ergodic which implies that it possesses a *steady-state probability distribution* π.

The synchronicity requirement for the state transitions in a PBN is an oversimplification of the real interactions that take place during genomic regulation. While it is not difficult to extend the PBN model into an asynchronous one, we do not discuss such extensions here. There are two reasons for focusing our attention to the synchronous case of PBN only. First, the model estimation from data is a much harder problem in the asynchronous case. Second, the synchronous PBN framework facilitates a simpler and clearer treatment of the problems about complexity-reducing mappings.

Originally, the PBN model was introduced by Shmulevich at al. (2002a). However the original definition concerned the *instantaneously random* PBN model only, i.e. the model where $p = 0$ and $q = 1$. The definition of context-sensitive PBN not only includes instantaneous PBNs as a special case but also allows for the interpretation of data obtained from distinct sources, each representing a specific cell *context*. Thus, one interprets data as obtained from a family of deterministic BN, and the PBN is viewed as a collection of BNs in which one constituent network governs the gene activity for a random period of time before another randomly selected deterministic BN takes over which might be in response to external stimulate or activity of latent variables.

The BN and the PBN model of GRN can easily be extended to the case where each gene is allowed more than two values depending on the quantization procedure applied to data. The ternary quantization, where the gene expression is grouped into up-regulated (+1), down-regulated (-1) and invariant (0), has been often used in the case of cDNA microarray expression data. Even such seemingly simple extension of the model presents us with a problem about the network complexity: the size of its state space S increases from 2^n to 3^n and the dimensions of transition matrix P_n change accordingly. The situation is even worse when more genes are included in the network because the size of the state space grows exponentially with the number of genes. Thus, the network compression or reduction becomes important task in the practical applications of both the BN and the PBN models. Shmulevich and Dougherty (2003a) were the first to consider various types of mappings between instantaneously random PBNs. The projection mapping proposed there aims at reducing the size of the model by deleting one gene from the set of network nodes. The mapping is designed in such a way that it preserves the probably structure of the original PBN. Here, following (Ivanov at al., 2007), we provide the definition of the basic projection mapping for the general case of context-sensitive PBNs. The basic projection Π_i is a mapping that transforms a given PBN A into a new one with the same parameters q and p, and such that the number of genes is reduced by one, i.e. the gene x_i in the original network is 'deleted'. Without loss of generality, we may assume that the deleted gene is x_n. Thus, for A

$\Pi_n : A \to \hat{A}_n$

$\hat{A}_n^{q,p}\left(\hat{V},\hat{F},\hat{C}\right), \hat{V} = \{x_1,...,x_{n-1}\}, \hat{F} = \{\hat{\boldsymbol{f}}_1,...,\hat{\boldsymbol{f}}_r\}, \hat{C} = (\hat{c}_1,...,\hat{c}_r)$

Every predictor function $\boldsymbol{f}_j$, i=1,...,n-1, generates two predictors $\hat{\boldsymbol{f}}_{0j}$ and $\hat{\boldsymbol{f}}_{1j}$ according to the rule

$$\hat{\boldsymbol{f}}_{kj}\left(x_1,...,x_{n-1}\right) = \boldsymbol{f}_j\left(x_1,...,x_{n-1},k\right), k \in \{0,1\}, \forall (x_1,...,x_{n-1}) \quad (1)$$

Thus, every network function for A determines 2^{n-1} new network functions for $\hat{A}_n$ by combining the first n-1components of $\hat{\boldsymbol{f}}_{0j}$'s and $\hat{\boldsymbol{f}}_{1j}$'s in all of the possible ways for every fixed j. The new network functions have their corresponding selection probabilities given by the formula

$$c_j(\Pr\{x_n = 1\})^l(\Pr\{x_n = 0\})^{n-i-1}, j = 1,...,r \quad (2)$$

where l is the number of the components of the new network function that are coming from $\hat{\boldsymbol{f}}_{1j}$, and $\Pr\{x_n=k\}$, $k \in \{0,1\}$ is the marginal probability for the gene x_n to have values 0 or 1, computed using the steady/stationary state probability distribution of the original PBN A. For example, the new network function $\left[\hat{\boldsymbol{f}}_{11}^{(1)}, \hat{\boldsymbol{f}}_{01}^{(2)},..., \hat{\boldsymbol{f}}_{01}^{(n-1)}\right]$, where the upper indexes indicate the corresponding component of the vector, has its selection probability equal to $c_1(\Pr\{x_n = 1\})^l(\Pr\{x_n = 0\})^{n-1}$. One can see that in the process of 'deleting' a gene every predictor function that had that gene as an essential variable is replaced by two predictor functions in the new network. These two predictors capture the differences in the state transitions corresponding to the two different possible values for the gene that becomes a latent variable after the projection. When two or more of the network functions for $\hat{A}_n$ happen to be identical their selection probabilities combine in a natural way. The basic projection mapping Π_i can be repeatedly applied to achieve the desired reduction in the number of genes. At the same time, it has an obvious draw-back: the number of genes in a PBN is reduced at the expense of an exponential increase of the number of constituent BNs for the new PBN. Thus the projection mapping serves as an important example of the phenomenon that 'deleting' a gene from the network model can lead to a new model of higher complexity compared to the original one. Moreover, the projection mapping suggests that size-reduction mappings for models of GRNs should be defined as one-to-many mappings. It also provides us with an example of a compression approach for reduction of the complexity of a PBN that has already been inferred from data. A different approach is to penalize the network complexity during the process of inferring it from data. An example of such an approach, motivated by the MDL principle can be found in (Tabus at al., 2003) where the inferred network cannot be too complex because it is designed to have minimal stochastic complexity. The MDL principle states that, given a set of data and a class of models, one should choose the model that provides the shortest encoding of the data (Rissanen, 2007). From the perspective of inference, the MDL principle represents a form of complexity regularization, and in essence balances the deviation from data and the model complexity.

In the next section, we focus on the definition of the *reduction problem* for BN and PBN models of GRN, and then provide the reader with a detailed overview of different size-reduction mappings. Then, the MDL principle is used to address the question of estimating the cost of applying such mappings.

COMPLEXITY-REDUCING MAPPINGS

1. Constraints and Reduction

To better understand the role of constraints in the process of designing reduction mappings for PBN we focus on the Boolean networks which are the building blocks of a PBN. Consider the space M_n of all BNs on *n* genes. Then, having as an example the multi-valued basic projection mapping Π_i, a size-reducing mapping can be defined as any set valued mapping $\pi : M_n \to 2^{M_{n-1}}$, where $2^{M_{n-1}}$ denotes the set of all subsets of the space of Boolean networks on *n*-1 genes. Such a general definition takes into account only the 'deletion' of one of the genes from the networks in M_n, and is of little practical use. On the other hand side, it helps in formulating the following

Reduction Problem: Given a set of constraints Λ and a BN $B \in M_n$ find a reduction mapping $\pi : M_n \to 2^{M_{n-1}}$, such that every $\tilde{B} \in \pi(B)$ satisfies Λ.

Here we adopt a very general definition of a constraint for Λ, namely: a condition that should be satisfied by every solution of the stated problem. Thus, Λ partitions the space $2^{M_{n-1}}$ and allows for optimization procedures to be applied in finding the reduction mapping π.

Keeping in mind that the deterministic Boolean networks are the building blocks of the stochastic PBN model, it is straightforward to state the reduction problem for the case of a PBN. One should keep in mind that the constraints Λ could be internal with respect to the model, i.e. related to the dynamical or static structure of $\tilde{B}$ (the graphs $\tilde{\Gamma}$ or $\tilde{G}$) or *B* (the graphs Γ or *G*), or could be external with respect to the model. For example, Λ could be related to qualitative knowledge/description of the biological phenomena being modeled.

Several observations are worth mentioning.

- Given a set of constraints Λ and a BN $B \in M_n$ the problem of designing π can be interpreted as a constrained search problem where the search space is the direct product $\times_{i=1}^{n-1} T_i$ of truth tables T_i for the Boolean functions on *n*-1 variables. The set of constraints Λ helps to determine some of the entries in those truth tables which could significantly reduce the size of the search space. This interpretation of the reduction problem allows for algorithms that are used to design BNs from data, e.g. (Pal at al., 2005), to be used in determining the set $\pi(B)$. Conversely, complexity-reducing mappings can be used in designing models of GRNs, e.g. (Ivanov at al., 2006).
- Given a set of constraints Λ and a reduction mapping π there exists a maximal, with respect to the partial order induced by set inclusion, subset $\Omega_{\Lambda,\pi} \subseteq M_n$, such that the same reduction mapping π solves the reduction problem for every $\tilde{B} \in \Omega_{\Lambda,\pi}$ i.e. $\forall B \in \Omega_{\Lambda,\pi}$ all $\tilde{B} \in \pi(B)$ satisfy Λ.
- One should notice that there is a partial order induced by set inclusion for the sets of constraints. If π is a solution to the reduction problem for a given Λ_1 and $B \in M_n$ then if $\Lambda_1 \subset \Lambda_2$ π might not be anymore a solution to the reduction problem for Λ_2 and *B*. Thus, one can look for the maximal, with respect to this partial order, set of constraints Λ with $\Lambda_1 \subset \Lambda$, so that π solves the reduction problem for Λ and *B*.
- Given that one of the main reasons for constructing reduction mappings is to reduce the complexity of a network model, one can see that the choice of constraints Λ has a significant impact on achieving this goal. The cardinality of the set $\pi(B)$ could be so big that the mapping π leads

to an increase of the model complexity, as the example of the basic projection mapping shows. Moreover, the verification if a network from $\pi(B)$ satisfies Λ can be computationally intensive for some sets of constraints. For example, if Λ = {BN with singleton attractors only} then such a verification might require finding all of the attractor cycles for a BN $\in \pi(B)$ - a problem known to be NP-complete.

The basic projection mapping Π_n is a solution of the reduction problem for the set of constraints $\Lambda = \{\tilde{\boldsymbol{f}}^{(i)} = \hat{\boldsymbol{f}}_0^{(i)} \text{or } \hat{\boldsymbol{f}}_1^{(i)},\ i = 1,...,n-1\}$, where $\boldsymbol{f}$ is the network function for the BN to be reduced. As mentioned earlier, the cardinality of the set $\Pi_n(B)$ can be very large which leads to a significant complexity increase. The complexity increase is even worse when the basic projection mapping is applied to 'delete' a gene from a PBN, Eqs. (1), (2). On the other hand side, the projection mapping has some advantages. First, because Λ prescribes all of the entries in the truth table of each $\tilde{B}$ in terms of the predictor functions for $B, \Omega_{\Lambda,\Pi_n} \equiv M_n$ i.e. the projection can be applied to every BN on *n* genes. Second, for the same reason, there is no need to verify that every BN$\in \Pi_n(B)$ satisfies the constraint.

To remedy the basic projection mapping's problem of possible exponential increase of the constituent networks after a 'deletion' of one gene from the PBN, Ivanov and Dougherty (2004) proposed a new class of complexity-reducing mappings for PBNs. The mappings from that class come as solutions of a specific optimization problem, and do not increase the number of the contexts of the original PBN. However, these mappings might fail to preserve the probability structure of that PBN. Originally, the reduction mappings were defined for the case of instantaneously random PBNs. The following discussion which treats the general case of context-sensitive PBNs is borrowed from (Ivanov at al., 2007).

To better understand the motivation and the definition of the reduction mapping, we consider a PBN $A = A^{q,p}(V,\boldsymbol{F}, C)$ and the following portion of its transition matrix P_n containing the transition probabilities for the states $\boldsymbol{s}_1 = (x_1,...,x_{n-1},1)$, $\boldsymbol{s}_0 = (x_1,...,x_{n-1},0)$, $\boldsymbol{s}'_1 = (x'_1,...,x'_{n-1},1)$ and $\boldsymbol{s}'_0 = (x'_1,...,x'_{n-1},0)$.

$$\begin{array}{ccc} & \boldsymbol{s}'_1 \cdots & \boldsymbol{s}'_0 \\ \boldsymbol{s}_1 & p_{\boldsymbol{s}_1\boldsymbol{s}'_1} \cdots & p_{\boldsymbol{s}_1\boldsymbol{s}'_0} \\ \vdots & \vdots & \vdots \\ \boldsymbol{s}_0 & p_{\boldsymbol{s}_0\boldsymbol{s}'_1} \cdots & p_{\boldsymbol{s}_0\boldsymbol{s}'_0} \end{array}$$

If one 'deletes' the gene x_n these four transitions collapse to one transition $\boldsymbol{s} \xrightarrow{p^*_{\boldsymbol{s}\boldsymbol{s}'}} \boldsymbol{s}'$ where $\boldsymbol{s} = (x_1,...,x_{n-1})$, $\boldsymbol{s}' = (x'_1,...,x'_{n-1})$ and

$$p^*_{\boldsymbol{s}\boldsymbol{s}'} = \Pr\{x_n = 1\}(p_{\boldsymbol{s}_1\boldsymbol{s}'_1} + p_{\boldsymbol{s}_1\boldsymbol{s}'_0}) + \Pr\{x_n = 0\}(p_{\boldsymbol{s}_0\boldsymbol{s}'_1} + p_{\boldsymbol{s}_0\boldsymbol{s}'_0}) \tag{3}$$

Here, just as in the case of the basic projection mapping, $\Pr\{x_n=k\}$, $k \in \{0,1\}$ is the marginal probability for the gene x_n to have values 0 or 1, computed using the steady/stationary state probability distribution of the original PBN *A*. Now, if one considers the reduction problem for the constraint Λ_1 = {keep the number of contexts/constituent BNs and the list of the selection probabilities *C* unchanged} and the network *A*, one can notice that in order to optimally preserve the probability structure of the network, e.g. the basic projection mapping, an additional constraint has to be imposed, namely, $\Lambda_\varepsilon = \{\tilde{p}_{\boldsymbol{ij}} : \left|\tilde{p}_{\boldsymbol{ij}} - p^*_{\boldsymbol{ij}}\right| \le \varepsilon\ \forall \boldsymbol{i}, \boldsymbol{j} - \text{states} \in \Gamma\}$ where $0 \le \varepsilon \le 1$ and $\tilde{p}_{\boldsymbol{ij}}$ denotes the corresponding transition

probability in the reduced network $\tilde{A}$. Thus, the reduction problem should be considered with respect to the constraint $\Lambda_1 \cup \Lambda_\varepsilon$. The introduction of the constraint Λ_ε is natural, given that the basic projection mapping preserves the probability structure of a PBN perfectly and could be considered as a bench mark for evaluating the performance of other complexity-reducing mappings when the preservation of the probability structure of the model is one of the main objectives of the reduction. It is clear that the choice of ε is important if one wants to have a solution to this reduction problem. Moreover, ε might depend on the PBN *A* and this could potentially complicate the procedure of finding the correct ε for a given PBN. Ivanov and Dougherty (2004) avoided these complications by re-casting the constraint Λ_ε as an optimization problem over a compact set. Therefore, for any given PBN one is ensured to have a solution to the reduction problem with the smallest possible ε. The optimization is based on a procedure that combines $\hat{\boldsymbol{f}}_{0j}$ and $\hat{\boldsymbol{f}}_{1j}$, j =1,…, r to form the new network function $\tilde{\boldsymbol{f}}_j$. One should note an important difference between the basic projection mapping and the reduction mapping. While the projection is based on the marginal probability distribution of a single gene, the reduction mapping is defined using the probability distribution of the entire collection of states of *A* (Ivanov, 2004). In both cases though, there is no control over the changes in the dynamics/state transition diagrams of the BNs comprising the original PBN. In addition, both mappings rely on knowledge about the steady/stationary state distribution of the original PBN.

Finding a minimal set of constraints presents us with yet another problem because there is only a partial order within the collection of all sets of constraints. Seemingly natural constraints could lead to significant changes in either the structure or the dynamics of the reduced PBN as the following discussion shows.

One such a constraint arises if we consider the local properties of the predictor functions, and is based on the following.

Definition 3: Given a state $\boldsymbol{s} \in \Gamma$, the predictor function f^i is called $(\boldsymbol{s}, j)$ independent if it has partial derivative $\dfrac{\partial f^i}{\partial x_j}(s) = 0$.

The $(\boldsymbol{s}, j)$ independence is a local property of a predictor function and suggests that if a toggle of x_j in state $\boldsymbol{s}$ does not affect the prediction of gene x_i then the new predictor function $\tilde{f}^i$ in any of the reduced networks $\tilde{B}$ should have the same value as $f^i(\boldsymbol{s})$ at the state $\tilde{\boldsymbol{s}}$ that is obtained from $\boldsymbol{s}$ by 'deleting' the *j*-th gene. Thus, we arrive at the local constraint $\Lambda_2 = \{\forall(\boldsymbol{s}, j)$ independent $f^i, \boldsymbol{s} \in \Gamma, I, j = 1,\ldots, n\text{-}1$ define $\tilde{f}^i(\tilde{\boldsymbol{s}}) = f^i(\boldsymbol{s})\}$.

Another constraint Λ_3 is also related to a local property but this time it is about the digraph of gene dependencies *G*. If one interprets a 'deleted' gene x_j as a latent variable, then any two edges $x_k \to x_j$ and $x_j \to x_i$ in *G* should produce an edge $x_k \to x_i$ in $\tilde{G}$ for any of the reduced networks $\tilde{B}$. Thus, if the *j*-th gene is 'deleted' $\Lambda_3 = \{\forall(x_k \to x_j, x_j \to x_i)$ define the new predictor set for the gene x_i by first removing x_j from its original predictor set and then adding the gene x_k as one of its predictor genes, i.e. $\tilde{W}_i = \{W_i \setminus \{x_j\}\} \cup \{x_k\}\}$.

One can combine Λ_2 and Λ_3 into a new constraint $\Lambda_2 \cup \Lambda_3$ and then look for a solution to the reduction problem with respect to this constraint. The following example shows that although $\Lambda_2 \cup \Lambda_3$ is biologically meaningful, it has little to do with the global dynamical properties of the state transition diagram Γ of the Boolean network.

Table 1. Truth table for B

$x_1x_2x_3$	f^1	f^2	f^3
0 0 0	0	0	0
0 0 1	1	1	0
0 1 0	1	1	0
0 1 1	1	0	1
1 0 0	0	1	0
1 0 1	1	1	1
1 1 0	1	0	1
1 1 1	1	1	1

Example 1: Consider the reduction problem for the constraint $\Lambda_2 \cup \Lambda_3$ and the Boolean network $B \in M_3$ given by the truth table, Table 1:

One can easily check that there are 10 networks in the set $\pi(B)$ and all of those networks satisfy the constraint $\Lambda_2 \cup \Lambda_3$. At the same time, while the state transition diagram for *B* has two singleton attractors, only one of those 10 networks has such a property.

This example points out to the importance of constraints related to the global dynamical properties of the networks. In a recent attempt to address this problem Ivanov at al. (2007) used the state transition diagram itself as a constraint for designing complexity-reducing mappings. The idea behind the *Dynamics Induced Reduction (DIRE)* algorithm developed in that paper is based on two important interpretations of the process of removing/'deleting' a gene from a PBN.

1. First, it is desirable that deleting a gene from a given regulatory network does not increase the number of constituent BNs.
2. Second, it is important that the dynamics of a real genetic regulatory system do not strongly depend on how many of the genes participating in the regulation are observable or not. Thus, preserving properties of a PBN, such as its attractor structure, the relative sizes of the basins of attraction, as well as the level structure of the state transition diagrams of the BNs comprising the PBN, should be a major goal when designing complexity-reducing mappings.

DIRE collapses the state transition diagram of each one of the BNs that form the PBN in a manner similar to the example given by Figure 1. To illustrate the importance of the state transition diagram as a constraint we consider the BN from Example 1. Suppose that the gene corresponding to the right-most digit in the binary representation of the states is to be deleted. If we try to collapse the state transition diagram, we notice that, with respect to the attractor structure of the original BN, merging the node (001) and the attractor node (000) can be done in two very different ways: either the merging happens towards the attractor state or it happens towards the transient state. In the first case, the attractor state is preserved in the reduced BN as (00), and the basin of attraction of the attractor (111) in the original network looses one state. In the second case, the attractor structure of the reduced network differs significantly to that of the original BN, the only remaining attractor being the reduced state (11). Thus, if we consider the attractor structure of a BN as a representation of important biological characteristics

of the real GRN, then merging of the states (000) and (001) should be done towards the attractor state. At the same time, we point out that those two states are the only states in the original state transition diagram that create a possibility of essentially altering the attractor structure of the original BN. The rest of the states will merge within the basin of attraction of the attractor state (111). This simple example illustrates the importance of the concept of an *inconsistency point*.

Definition 4: A state $\boldsymbol{s}$ is called an *inconsistency point* with respect to gene x_i if and only if the state $\boldsymbol{s}'$ in the state transition diagram that differs from $\boldsymbol{s}$ only in the value x_i of belongs to a different basin of attraction compared to the basin of attraction that contains $\boldsymbol{s}$. The states $\boldsymbol{s}$ and $\boldsymbol{s}'$ are called *dual* with respect to the gene x_i.

In essence, DIRE attempts to preserve the flow of information between states lying on the same path of the state transition diagram, and it handles the exceptions created by the inconsistencies arising from deletion of a gene by considering it to be a latent variable. The inconsistency points are treated in a way that controls the damage to the state transition diagram in terms of attractors and their basins. The probabilistic parameters *p*, *q* and *C* of the original PBN are preserved, and the number of the network contexts is essentially unchanged (Ivanov at al., 2007).

DIRE uses the state transition diagrams of the PBN contexts as a constraint and thus preserves in an optimal sense the dynamical structure of the network. At the same time, the structural dependencies among the genes are not taken into a consideration. Applying the algorithm to the BN from Example 1 produces a network that does not possess the dependency $x_2 \rightarrow x_2$ that is present in the digraph *G* of *B*. Such a dependency could represent important feedback which is part of the real GRN and it might be desirable to keep it in the reduced network. Thus, in some cases, the entire state transition diagram could be too strong of a constraint in solving the corresponding reduction problem.

2. Criteria for Evaluating Complexity-Reducing Mappings

There are several criteria that could be used when comparing different complexity-reducing mappings. These criteria relate to properties of either the model or the real GRN, and can be used to produce constraints for the reduction problem. The five major comparison criteria are:

i. Structural inter-gene relationships, e.g. the digraphs *G* for the different contexts of a PBN or the number of those contexts.
ii. Dynamical properties of the model, e.g. the state transition diagrams Γ for those contexts or the steady/stationary state distribution of the model.
iii. Biology related, e.g. types of attractors (singleton or cyclic) and their lengths.
iv. The complexity of the reduction mapping itself, both computational and stochastic.
v. The performance of control and intervention strategies when combined with complexity-reducing mappings.

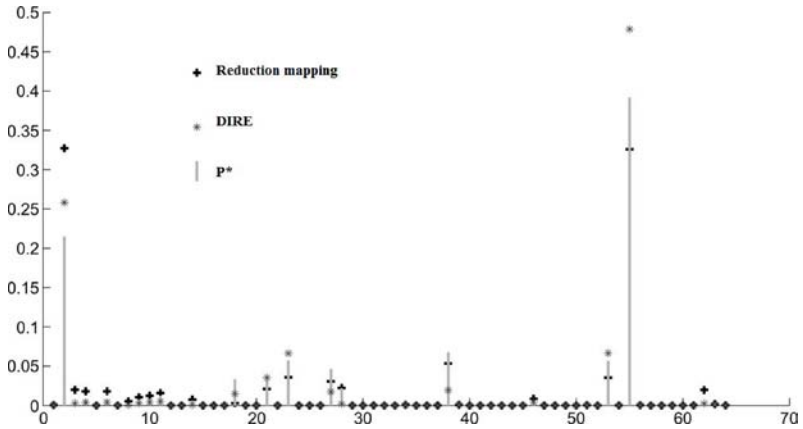
As the main objective of the modeling is to predict the dynamical behavior of the system and ultimately provide the effective intervention strategies for prevention and control of disease that is associated with altered genomic regulation, the most often used criteria for comparison of complexity-reduction mappings are the steady-state probability distribution of the network, its attractor structure, and the performance of control policies designed for network intervention.

The three different kinds of reduction mappings considered so far, illustrate that, in the case of the PBN model, it is very difficult to find a complexity-reducing mapping that is optimal with respect of all of these criteria. The basic projection mapping preserves the probability structure of the PBN but is definitely the worst with respect to the criterion. On the other hand side, the basic reduction mapping is not computationally expensive. DIRE minimizes the damage, induced by a gene 'deletion', to the state-transition diagram of the network and performs very well with respect to the first criterion but as our example shows might not preserve structural inter-gene dependencies. In addition, it could be computationally prohibitive as it requires traversing the state-transition diagram. The reduction mapping also performs well with respect to the criterion i. and is similar to DIRE, although DIRE performs slightly better, when one uses the steady-state distribution as a measure of the mapping's performance. This holds not only for synthetic data but also when one reduces PBN models of real data as the example borrowed from (Ivanov at al., 2007) shows. For brevity sake we do not discuss the method that was used to infer the network used in this example. Interested readers should check out the details given in (Zhou at al., 2004). The main objective here is to provide an example of how the above stated criteria can be used to compare and evaluate the performance of mappings that solve the reduction problem for different sets of constraints.

Example 2 (Melanoma application): The binarized gene-expression profiles used in the study result from data from 31 malignant melanoma samples (Bitner at al., 2000). The seven genes WNT5A, pirin, S100P, RET1, MART1, HADHB, and STC2 used here for the model were chosen from a set of 587 genes from the melanoma data set that were subjected to an analysis of their ability to cross predict each other's state in a multivariate setting (Kim at al., 2002). A PBN comprised of four BNs was constructed from the binarized gene-expression profiles following the method from (Zhou at al., 2004). The parameters for the PBN A were set as $p = 0.01$ and $q = 0.01$. Figure 1 shows the steady-state distributions of the PBNs $\tilde{A}$ and $\widehat{A}$ produced by deleting WNT5A using the reduction mapping and DIRE, respectively, together with the probability distribution p^* defined on the same state space and resulting from the collapsing procedure given in Eq. (1.3). The *Mean Square Error* (MSE) between p^* and the steady-state distribution of $\widehat{A}$ is 0.01369498, whereas the MSE between p^* and the steady-state distribution of $\tilde{A}$ is 0.0242964.

Estimating the computational complexity of reduction mappings could be considered a routine procedure as long as one can describe an algorithm that produces the corresponding reduced networks. Such estimates are usually related to the size of the sets $\pi(B)$ for every context B of the given PBN and the mapping π. However, the question about how the cost of applying such mappings could be measured has been largely ignored. Dougherty and Ivanov (2008) were the first to propose to use the notion of the stochastic complexity to measure the cost of reduction for a special case of PBNs, BNs with perturbation. Approaching the problem of estimating the cost of reduction from such a perspective is natural because every PBN can be viewed as data generating machinery. This view is also consistent with information-theoretic considerations based on the MDL principle and the NML model. The cost of reduction is given by the relative change in the stochastic complexity of the network after applying a reduction mapping. The simulations show that there is a clear correlation between the relative change in the stochastic complexity and the ℓ_1 distance between the steady-state distributions of the original and the reduced networks. In addition, there is a large variance in the relative change in the stochastic complexity for some of the studied networks which is due to 'deleting' different genes from it. Thus,

Figure 1. Steady-state distributions



the proposed method for measuring the cost of reduction has the potential to rank genes in the network with respect to their impact on the steady-state distribution.

The effects of complexity-reducing mappings on the performance of various control policies, e.g. v., are largely unknown. Ghaffari at al. (2008) were the first to study how the *Mean First Passage Time* (MFPT) (Vahedi, Faryabi at al., 2008) control policy designed on the reduced network could be extended to the original network in the special case of BNs with perturbation. To address the issue of changing the long-run behavior, stochastic control has been employed to find stationary control policies that affect the steady-state distribution of a PBN. The algorithms used to find these solutions have complexity which increases exponentially with the number of the genes in the network. Hence, there is a need to study how size-reducing mappings could be used to produce new and more tractable models whose stationary control policies induce sub-optimal stationary control policies on the larger PBN. The results suggest that in the case of BN with perturbation one can use the control policy designed on the reduced network to approximate the control policy for the original model. The approximation fails only for intervention policies where the cost of applying control is small.

CONCLUSION

The inherent high complexity of the network models of genomic regulation creates the need for mappings that reduce the size of the model while preserving its biologically meaningful properties. There are many different definitions of complexity, and we elect to focus mainly on the one that is commonly accepted in systems theory where complex systems are characterized by the presence of many interacting parts and the interactions are generally non-linear and stochastic. We also describe the initial steps in applying methods from the theory of algorithmic complexity to measure the Kolmogorov complexity of both the network model and the reduction mapping. The research in this direction is still in its infancy but the MDL principle seems to provide the necessary framework for advances in the future. Using the Boolean network model and its stochastic generalization, the probabilistic Boolean network model, as examples we formulate the general reduction problem. The examples of complexity-reduction mappings demonstrate that there is no 'ultimate' solution to that problem. Rather, depending on the set of design

constraints there could be many or none candidate networks that could be considered as reductions of the original one, a situation which is common for ill-posed inverse problems. The number of possible solutions of the reduction problem depends on the constraints used to narrow down the size of the search space. We show the importance of the local and global properties of the inter-gene dependencies digraph and the state-transition diagram of a PBN in building up a set of constraints that not only reduces the size of the search space but also has proper biological interpretation. The interpretation of the reduction problem as a search problem provides a new perspective when considering the problem about network inference from data. Specifically, understanding how different sets of constraints affect the reduction helps to impose proper constraints when inferring the model from data. We propose several important criteria (i. - v.) for evaluation of complexity-reducing mappings, and give examples of their applications.

FUTURE RESEARCH DIRECTIONS

Mathematical modeling of genomic regulation has the potential to unravel the mechanisms of cell functioning from a systemic perspective. There are several important goals of the modeling process:

- To infer optimal models from data.
- To characterize the dynamical behavior of the real GRN in terms of steady-state probability distributions of the model and the structure of its attractors and their basins.
- To investigate how structure determines dynamics and the restrictions imposed by a specific dynamical behavior on the structure of the GRN.
- To characterize possible intervention strategies for control of the dynamical behavior of the system.

Complexity-reducing mappings for network models of genomic regulation can serve not only as a tool to produce smaller and more tractable models. The careful study of this ill-posed inverse problem can provide insights to model inference, constraints on the structure and dynamics of the network, and most importantly on the possibility for designing effective intervention strategies which control the dynamical behavior of the real GRN. As the genomic regulatory system is highly complex one has to deal with highly complex models as well and reduction mappings could be of great help. However, if the reduction mapping itself is highly complex, either in terms of algorithmic or stochastic complexity, such mapping is of little practical use. Thus, one has to address satisfactory the following questions:

- Develop a sound methodology for complexity estimation of both the model and the mappings used to reduce its complexity.
- Study how intervention strategies designed for the reduced networks perform on the larger ones. This in its turn requires investigation of the relationship between the concepts of controllability and reducibility of the models.
- Establishing various minimal sets of constraints that are optimal for solving the reduction problem depending on the practical application of the model.

These important future research directions represent some of the emerging trends in the field of mathematical and computational modeling of GRNs.

REFERENCES

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., & Hendrix, M. (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature, 406*(6795), 536–540. doi:10.1038/35020115

Codd, E. F. (1968). *Cellular automata.* New York: Academic Press.

Crick, F. (1970). Central dogma of molecular biology. *Nature, 227*, 561–563. doi:10.1038/227561a0

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*, 67–103. doi:10.1089/10665270252833208

Dougherty, E. R., & Braga-Neto, U. (2006). Epistemology of computational biology: Mathematical models and experimental prediction as the basis of their validity. *Journal of Computational Biology, 14*(1), 65–90.

Dougherty, E. R., Datta, A., & Sima, C. (2005). Research issues in genomic signal processing. *IEEE Signal Processing Magazine*, *22*(6), 46–68. doi:10.1109/MSP.2005.1550189

Dougherty, J., & Ivanov, I. (in press). Reduction cost for Boolean networks with perturbation.

Ghaffari, N., Ivanov, I., & Dougherty, E. R. (in press). Reduction mappings and control policies for intervention in Boolean networks.

Glass, K., & Kauffman, S. (1973). The logical analysis of continuous, nonlinear biochemical control networks. *Journal of Theoretical Biology, 39*, 103–129. doi:10.1016/0022-5193(73)90208-7

Huang, S. (1999). Gene expression profiling, genetic networks and cellular states: An integrating concept for tumorigenesis and drug discovery. *Molecular Medicine (Cambridge, Mass.), 77*(6), 469–480.

Huang, S. (2001). Genomics, complexity and drug discovery: Insights from Boolean network models of cellular regulation. *Pharmacomgenomics*, *2*(3), 203–222. doi:10.1517/14622416.2.3.203

Ivanov, I., & Dougherty, E. R. (2004). Reduction mappings between probabilistic Boolean networks. *EURASIP Journal on Applied Signal Processing, 1*, 125–131. doi:10.1155/S1110865704309182

Ivanov, I., Pal, R., & Dougherty, E. R. (2006), Applying reduction mappings in designing genomic regulatory networks. *IEEE/NLM Life Science Systems and Applications Workshop*, 1–2.

Ivanov, I., Pal, R., & Dougherty, E. R. (2007). Dynamics preserving size reduction mappings for probabilistic Boolean networks. *IEEE Transactions on Signal Processing, 55*(5), 2310–2322. doi:10.1109/TSP.2006.890929

Kauffman, S. (1969a). Metabolic stability and epigenesis in randomly generated genetic nets. *Journal of Theoretical Biology*, *22*, 437–367. doi:10.1016/0022-5193(69)90015-0

Kauffman, S. (1969b). Homeostasis and differentiation in random genetic control networks. *Nature, 224*, 177–178. doi:10.1038/224177a0

Kauffman, S. (1973). The large scale structure and dynamics of genetic control circuits: An ensemble approach. *Journal of Theoretical Biology, 44*, 167–190. doi:10.1016/S0022-5193(74)80037-8

Kauffman, S. (1993). *The origins of order: Self-organization and selection in evolution.* New York: Oxford University Press.

Kim, S., Li, H., Chen, Y., Cao, N., Dougherty, E. R., Bittner, M. L., & Suh, E. B. (2002). Can Markov chain models mimic biological regulation? *Journal of Biological System, 10*, 337–357. doi:10.1142/S0218339002000676

Lähdesmäki, H., Hautaniemi, S., Shmulevich, I., & Yli-Hara, O. (2006). Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Processing*, *86*(4), 814–834. doi:10.1016/j.sigpro.2005.06.008

Mitchell, M., Crutchfield, J. P., & Hraber, P. T. (1994). Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D. Nonlinear Phenomena*, *75*, 361–391. doi:10.1016/0167-2789(94)90293-3

Pal, R., Data, A., Fornace, A. J., Bittner, M. L., & Dougherty, E. R. (2005). Boolean relationships among genes responsive to ionizing radiation in the NCI 60 ACDS. *Bioinformatics (Oxford, England), 21*(8), 1542–1549. doi:10.1093/bioinformatics/bti214

Pal, R., Ivanov, I., & Dougherty, E. R. (2005). Generating Boolean networks with a prescribed attractor structure. *Bioinformatics (Oxford, England), 21*(21), 4021–4025. doi:10.1093/bioinformatics/bti664

Rissanen, J. (2007). *Information and complexity in statistical modeling.* New York: Springer.

Shmulevich, I., & Dougherty, E. R. (2003a). Mappings between probabilistic Boolean networks. *Signal Processing*, *83*(4), 799–809. doi:10.1016/S0165-1684(02)00480-2

Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002a). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics (Oxford, England), 18*(2), 261–274. doi:10.1093/bioinformatics/18.2.261

Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002b). From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE, 90*, 1778–1792. doi:10.1109/JPROC.2002.804686

Shmulevich, I., Lähdesmäki, H., Dougherty, E. R., Astola, J., & Zhang, W. (2003b). The role of certain post classes in Boolean network models of genetic networks. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(19), 10734–10739. doi:10.1073/pnas.1534782100

Somogyi, R., & Sniegoski, C. (1996). Modeling the complexity of gene networks: Understanding multigenic and pleiotropic regulation. *Complexity*, *1*, 45–63.

Tabus, I., & Astola, J. (2001). On the use of the MDL principle in gene expression prediction. *Journal of Applied Signal Processing*, *4*, 297–303. doi:10.1155/S1110865701000270

Tabus, I., Rissanen, J., & Astola, J. (2002). Normalized maximum likelihood models for Boolean regression with application to prediction and classification in genomics. In W. Zhang & I. Shmulevich (Eds.), *Computational and statistical approaches to genomics*. Boston, MA: Kluver.

Tabus, I., Rissanen, J., & Astola, J. (2003). Classification and feature gene selection using the normalized maximum likelihood model for discrete regression. *Signal Processing, 84*(4), 713–727. doi:10.1016/S0165-1684(02)00470-X

Vahedi, G., Faryabi, B., Chamberland, J.-F., Data, A., & Dougherty, E. R. (in press). Intervention in gene regulatory networks via a stationary mean-first-passage time control policy.

Vahedi, G., Ivanov, I., & Dougherty, E. R. (in press). Inference of Boolean networks under constraint on bidirectional gene relationships.

Zhao, W., Serpedin, E., & Dougherty, E. R. (2006). Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics (Oxford, England), 22*(17), 2129–2135. doi:10.1093/bioinformatics/btl364

Zhou, X., Wang, X., Pal, R., Ivanov, I., Bitner, M., & Dougherty, E. R. (2004). A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics (Oxford, England), 20*(17), 2918–2927. doi:10.1093/bioinformatics/bth318

KEY TERMS AND DEFINITIONS

Gene Regulatory Network: A network of relations among strands of DNA (genes) and the regulatory activities associated with those genes.

Complexity: Understood in the context of either complex system or algorithmic information theory.

Boolean Network (BN): A mathematical model that describes genomic regulation as a deterministic discrete dynamical system.

Probabilistic Boolean network (PBN): A mathematical model that describes genomic regulation as a stochastic discrete dynamical system.

Reduction Problem: The ill-posed inverse problem for reducing the size and the complexity of a given computational model of genomic regulation under a given set of constraints.

Projection Mapping: A mapping that solves the reduction problem for the PBN model of genomic regulation under a specific set of constraints.

Reduction Mapping: A mapping that solves the reduction problem for the PBN model of genomic regulation without increasing the number of the constituent BNs.

DIRE Algorithm: An algorithm that construct a mapping that solves the reduction problem using the state transition diagram of a given PBN as a constraint.

Cost of Reduction: A measure for evaluating the complexity of a given reduction mapping.

Chapter 15
Abstraction Methods for Analysis of Gene Regulatory Networks

Hiroyuki Kuwahara
Carnegie Mellon University, USA; Microsoft Research - University of Trento CoSBi, Italy

Chris J. Myers
University of Utah, USA

ABSTRACT

With advances in high throughput methods of data collection for gene regulatory networks, we are now in a position to face the challenge of elucidating how these genes coupled with environmental stimuli orchestrate the regulation of cell-level behaviors. Understanding the behavior of such complex systems is likely impossible to achieve with wet-lab experiments alone due to the amount and complexity of the data being collected. Therefore, it is essential to integrate the experimental work with efficient and accurate computational methods for analysis. Unfortunately, such analysis is complicated not only by the sheer size of the models of interest but also by the fact that gene regulatory networks often involve small molecular counts making discrete and stochastic analysis necessary. To address this problem, this chapter presents a model abstraction methodology which systematically performs various model abstractions to reduce the complexity of computational biochemical models resulting in substantial improvements in analysis time with limited loss in accuracy.

INTRODUCTION

Thanks to advances in technologies, in genetic regulatory networks—where, for instance, high-throughput gene expression analysis methods are available and a vast amount of quantitative data has been collected—the information required for building quantitative models of gene regulatory networks can be obtained. The most exact way to simulate a quantitative model of a molecular system is *molecular dynamics* where movements of every molecule in the system are tracked (Gillespie, 2005, 2007). The system state of molecular dynamics is the positions and velocities of every molecule in the system where the dynamics

DOI: 10.4018/978-1-60566-685-3.ch015

of the system state are described by capturing every movement and every collision of molecules in the system. While this approach can show the time evolution of species' populations as well as the spatial distribution of each species, acquiring such detailed knowledge and performing such computationally expensive simulations is typically infeasible. By making the well-stirred assumption, the spatial property of a system can be abstracted away, overriding the system state to be simply the populations of species in the system. While this assumption greatly simplifies the complexity of models, it adds uncertainty in the time evolution of the system owing to the insufficient knowledge of the system descriptions that the very assumption precludes.

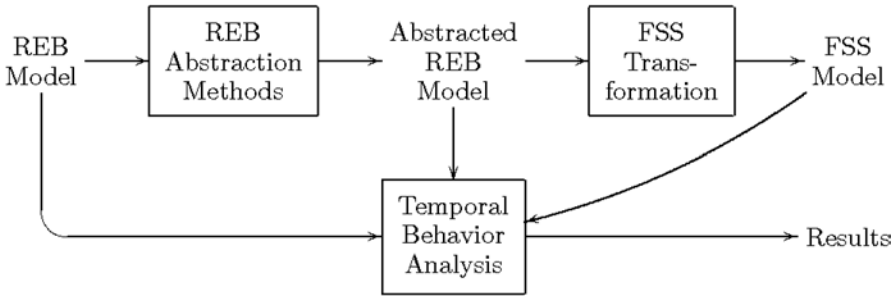
Traditionally, gene regulatory networks are modeled and analyzed within the continuous-deterministic, *classical chemical kinetics* (CCK) framework based on the *law of mass action* where the dynamics of a well-stirred system are described by a set of *ordinary differential equations* (ODEs). Although such treatment can be justified when the molecular populations are very large—and hence a CCK analysis may provide one of the most efficient approaches to estimate the time evolution of a system—the limitations of the CCK analysis have been broadly accepted (Arkin et al., 1998; Gillespie, 1992a, 2000; Elowitz et al., 2002; Rao et al., 2002; Samoilov et al., 2005; Samoilov and Arkin, 2006). In particular, given the same initial condition, the CCK analysis of biochemical systems always produces the same results as it neglects uncertainty in system dynamics. Furthermore, many regulatory components (e.g., DNA, RNA, proteins) in biological systems can be present in amounts too small to simply neglect the effects of inherent fluctuations (McAdams and Arkin, 1997; Golding et al., 2005; Raser and O'Shea, 2005; Pedraza and van Oudenaarden, 2005; Cai et al., 2006; Newman et al., 2006).

In order to more accurately predict the temporal behavior of gene regulatory networks, the *stochastic chemical kinetics* (SCK) framework can be used (Gillespie, 2005, 2007). Assuming that the system is spatially homogeneous, this SCK approach describes the time evolution of a biochemical system at the individual reaction level by exactly tracking the quantities of each molecular species and by treating each reaction as a separate random event. One consequence of SCK is a stochastic process description of the system that is analytically governed by the *chemical master equation* (CME) (McQuarrie, 1967; Gillespie, 1992b). However, directly obtaining the solution of the CME of any realistic system, either analytically or numerically, is not feasible due to its intrinsic complexity.

Instead of attempting to solve the CME, exact numerical realizations of a SCK model via Gillespie's *stochastic simulation algorithm* (SSA) (Gillespie, 1976, 1977), which is derived from the same premise as the CME, are often used to infer the temporal system behavior with a much smaller memory footprint. Unfortunately, the computational requirements of the SSA can be substantial due largely to the fact that it not only requires a potentially large number of simulation runs in order to estimate the system behavior at a reasonable degree of statistical confidence, but it also requires every single reaction event to be simulated one at a time.

Ultimately, given the substantial computational requirements of stochastic simulations, abstraction is absolutely essential for efficient computational analysis of complex gene regulatory networks. For such networks, any applications of the all-inclusive, low-level, quantitative models are largely impractical because of high computational demands, while the use of entirely high-level qualitative representations is typically inadequate owing to the substantial dynamical and functional complexity they can manifest. Therefore, a search for some intermediate level of abstraction becomes necessary. This, however, frequently presents a problem: while most abstractions used in modeling of biochemical networks have traditionally been implemented manually on a mechanism-by-mechanism basis, doing so accurately in general settings is a tedious and time-consuming process, which is highly susceptible to errors during model translation and transformation.

Figure 1. Automated model abstraction tool flow



To address the issues surrounding *in silico* analysis of biochemical systems, this chapter presents an automated model abstraction methodology of biochemical system descriptions based on chemical reaction kinetics (Kuwahara et al., 2006a; Kuwahara, 2007). This approach systematically reduces the small-scale complexity found in biochemical systems represented by *reaction-based* (REB) models (i.e., models composed of a set of chemical reactions) while broadly preserving the large-scale system behavior. Thus, this approach alleviates the abstraction problems by systematically testing network patterns and characteristics to determine which abstraction methods are applicable (Kuwahara et al., 2005, 2006a). Furthermore, this approach allows one to scan through the effective levels of abstraction and to optimize model transformation for *efficiency-versus-accuracy* by adjusting the various precision criteria for each abstraction method before its application.

Our methodology shown in Figure 1 begins with a REB model which could be simulated via the SSA or one of its variants though at a substantial computational cost. To reduce the cost of computational analysis, the original REB model is simplified by applying abstraction methods that mainly attempt to reduce the number of reactions and species based on the structure of the model and the abstraction criteria. The result is an abstracted REB model with fewer reactions and species, substantially lowering the cost of stochastic simulation. To further reduce the complexity of the system as well as analysis time, this abstracted REB model can be automatically translated into a *finite state system* (FSS) model by representing the dynamics of the system states (i.e., molecular population levels in the system) by a finite state graph. This model can then be efficiently analyzed, for example, using a Markov chain analysis method.

This chapter is organized as follows. Section 2 presents an overview of SCK to model and analyze gene regulatory networks. Section 3 presents REB abstraction methods. Section 4 defines the FSS model and a method to transform a REB model into a FSS model. Section 5 presents a case study of the automated model abstraction methodology. Finally, Sections 6 and 7 present a summary of the chapter and discussion for future directions.

BACKGROUND

Section 2.1 first presents a brief overview of gene regulatory networks. Section 2.2 presents an overview of SCK to computationally model and analyze such biochemical networks. The following subsections present various abstraction approaches to alleviate the computational costs of SCK analysis. Section 2.3

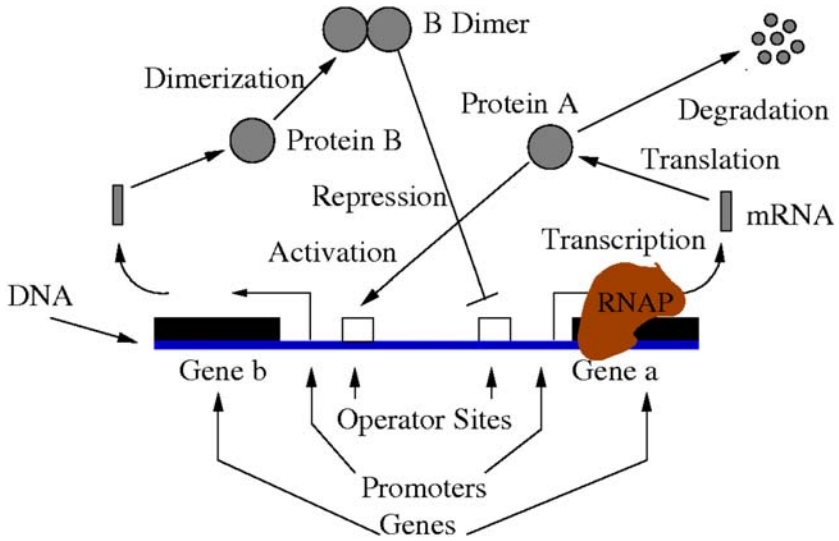
describes examples of abstraction in the simulation phase, while Section 2.4 describes examples of abstraction in the modeling phase.

Gene Regulatory Networks

Although gene expression can be regulated at each step of transcription and translation, the heart of gene regulation comes from the transcription initiation where *transcription factors* and *cis-regulatory* DNA elements control when and how genes are transcribed and in turn proteins are synthesized (DeRisi et al., 1997; Lodish et al., 1999; Causton et al., 2001; Davidson et al., 2002; Hermsen et al., 2006). Transcription factors are largely regulatory proteins that can control the rate of transcription by occupying *cis*-regulatory elements on DNA. Negative transcription factors called *repressors* prevent transcription of genes upon binding to the corresponding *cis*-regulatory elements, while positive transcription factors called *activators* enhance transcription of genes. *Cis*-regulatory elements on DNA include promoters and operators. Operators are segments of DNA that are usually located near the corresponding promoters of genes to which transcription factors can bind to repress or activate transcription. These critical transcription regulatory components can be present in very low counts in a cell (Guptasarma, 1995), contributing to the nondeterministic effects in gene expression (McAdams and Arkin, 1997; Elowitz et al., 2002; Samoilov and Arkin, 2006).

Figure 2 shows a relatively simple two-gene system to illustrate the mechanism of transcriptional regulatory networks. In this network, a piece of DNA contains two genes: *a* and *b*. Suppose proteins *A* and *B*, the products from genes *a* and *b*, are not present in the system, and the promoter for gene *a* has a higher affinity to RNAP binding than the promoter for gene *b* at the basal rate. Transcription of gene *a* is then initiated much more frequently than that of gene *b*, causing gene *a* to be expressed and protein *A* to be synthesized more often in this configuration. Protein *A* is an activator of transcription of gene *b* and it can occupy the operator site of gene *b* to increase the expression rate of gene *b*. Thus, protein

Figure 2. Two-gene system to illustrate the structure and the mechanism of transcriptional regulation. Protein A activates expression of gene b, while protein B represses expression of gene a.



B can be synthesized at a higher level. Two copies of protein *B* can *dimerize* and form a molecule, and this *Bdimer* can act as a repressor of transcription of gene *a* by occupying the operator site of gene *a*. Consequently, at high levels of protein *B*, protein *A* is rarely produced, and with degradation, the level of protien *A* becomes so low that protein *A* can no longer effectively occupy the operator site of gene *b* to activate the expression of gene *b*. Thus, with limited production at its basal rate, protein *B* degrades, allowing gene *a* to be expressed once again.

Stochastic Chemical Kinetics

In well-stirred chemical and biological molecular systems, including gene regulatory networks, REB representations typically provide the most detailed level of specification for the underlying system structure and dynamics (Berry et al., 2000). An REB model is composed of *N* chemical species $\mathbf{S} \equiv \{s_1,...,s_N\}$ which interact through *Mirreversible*, reactions $\mathbf{R} \equiv \{r_1,...,r_M\}$ inside a well-stirred, chemically reacting system with a constant volume Ω in thermal equilibrium at some constant temperature. A REB model can be encoded in an emerging standard, the *Systems Biology Markup Language* (SBML) (Finney and Hucka, 2003). Thus, REB models can be conveniently constructed using SBML-compliant modeling tools. The use of this standardized format has the advantage of allowing for easy exchange of computational models by researchers as well as the ability to analyze models by a variety of SBML-compliant analysis tools.

An REB model can describe the time evolution of the system within the discrete-stochastic framework in continuous time. Thus, by denoting $\mathbf{X}(t) \equiv (X_1(t),...,X_N(t))$ the system state vector that represents the number of molecules of each s_i, the evolution of $\mathbf{X}(t)$, given that $\mathbf{X}(t_0) = \mathbf{x_0}$ (for $t \geq t_0$) can be defined rigorously by the SCK framework. In the SCK framework, each reaction, r_j, is viewed as a discrete random event that changes the system state by $\mathbf{v_j} \equiv (v_{1j},...,v_{Nj})$, called the *state change vector*, whose *i*-th element, v_{ij}, specifies the change in X_i. Thus, given the system is in state $\mathbf{x} \equiv (x_1,...,x_N)$, the system jumps to state $\mathbf{x} + \mathbf{v_j}$ as a consequence of a single r_j reaction event. A species s_i that is consumed by a reaction r_j (i.e., $v_{ij} < 0$) is known as a *reactant* for the reaction. A species s_i that is produced by a reaction r_j (i.e., $v_{ij} > 0$) is known as a *product* for the reaction.

The time that the next reaction event r_j occurs is governed by the *propensity function*, $a_j(\mathbf{x})$, which is defined as follows:

$$a_j(\mathbf{x})dt \equiv \begin{array}{l}\text{the probabilitythat,given } \mathbf{X}(t) = \mathbf{x}\text{, reaction } r_j \text{ occurs} \\ \text{inside } \Omega \text{ in the next infinitesimal time interval } [t, t+dt)\end{array} \quad (1)$$

where the infinitesimal time dt is taken to be so small that at most one reaction event occurs within the interval. Note that a species that is neither a reactant or product, but it affects the value of the propensity function, a_j, is known as a modifier for reaction r_j. Since the state change vector and the propensity function are the basis of the discrete-stochastic description of the SCK framework, they may be said to be the fundamental premise of stochastic chemical kinetics (Gillespie, 2005). The propensity function of each reaction r_j is quantified by first defining a specific probability rate constant c_j such that:

$$c_j dt \equiv \text{the probability that a randomly chosen combination of reactant molecules of } r_j \text{ inside } \Omega \text{ at time } t \text{ will transform via } r_j \text{ within the next infinitesimal time } dt. \quad (2)$$

The strict requirement of SCK—as with CCK—is that each reaction r_j is a distinct instantaneous event, which is essentially viewed as an elementary reaction. This means that, strictly speaking, each reaction in a SCK model must be either a unimolecular reaction or a bimolecular reaction. Suppose reaction r_j is a unimolecular reaction and in the form $s_1 \xrightarrow{c_j} \cdots$. Then, from Definition 2, this propensity function is defined as $a_j(\mathbf{x}) = c_j x_1$. The value of c_j for a unimolecular reaction r_j turns out to be the same as the reaction rate constant from CCK. In contrast, suppose reaction r_j is a bimolecular reaction of the form $s_1 + s_2 \xrightarrow{c_j} \cdots$. Then, the propensity function takes the form of $c_j x_1 x_2$. In this case, $c_j \Omega$ is numerically the same as the corresponding rate constant k_j in CCK. If, however, bimolecular reaction r_j is a homogeneous dimerization reaction of the form $2s_1 \xrightarrow{c_j} \cdots$, the propensity function becomes $a_j(\mathbf{x}) = \frac{c_j}{2!} x_1(x_1 - 1)$ and the relationship between the specific probability rate constant and the classical reaction rate constant becomes $c_j = 2k_j / \Omega$. Also, if reaction r_j is a trimerization reaction $3s_1 \xrightarrow{c_j} \cdots$ and if it is assumed to be an elementary reaction, then the propensity function of reaction r_j becomes $a_j(\mathbf{x}) = \frac{c_j}{3!} x_1(x_1 - 1)(x_1 - 2)$ where $c_j = 3! k_j / \Omega^2$. In general, the propensity function for an *n*-merization reaction: $ns_1 \xrightarrow{c_j} \cdots$ becomes $a_j(\mathbf{x}) = \frac{c_j}{3!} \frac{x_1!}{(x_1 - n)!}$, and the relationship between and becomes $c_j = n! k_j / \Omega^{n-1}$. Thus, in homogeneous *n*-merization reactions, while the reaction rates via CCK are not equal to the corresponding propensity functions, they can approximate the propensity functions very well. This is especially true when the molecular counts are relatively high. This type of approximation is commonly applied to propensity functions, allowing biochemical system models to be numerically analyzed via both CCK and SCK approaches conveniently.

The time evolution of **X**(*t*) for an SCK model can be described by a temporally homogeneous jump Markov process that is described by the forward *chemical master equation* (CME):

$$\frac{\partial P(\mathbf{x}, t \mid \mathbf{x_0}, t_0)}{\partial t} = \sum_{j=1}^{M} [P(\mathbf{x} - \mathbf{v_j}, t \mid \mathbf{x_0}, t_0) a_j(\mathbf{x} - \mathbf{v_j}) - P(\mathbf{x}, t \mid \mathbf{x_0}, t_0) a_j(\mathbf{x})]. \quad (3)$$

Although the integral of the CME gives the probability $P(\mathbf{x}, t \mid \mathbf{x_0}, t_0)$ that captures the evolution of a biochemical system, directly obtaining the solution of the CME of most realistic systems, either analytically or numerically, is not feasible (Gillespie, 1977; van Kampen, 1992; Gardiner, 2004). This is because Equation 3 is actually a set of coupled, ordinary differential equations for each system state, and the system-state-space is usually very large, if not infinite, for realistic systems. Thus, owing to its intrinsic complexity, the CME itself is not particularly useful for analyzing the temporal behavior of biochemical systems.

In order to analyze an SCK model, Gillespie developed a Monte Carlo simulation algorithm called the *stochastic simulation algorithm* (SSA) (Gillespie, 1976, 1977). SSA is derived by defining a probability density function $p(\tau, j \mid \mathbf{x}, t)$ such that $p(\tau, j \mid \mathbf{x}, t) d\tau$ is the probability that, given $\mathbf{X}(t)=\mathbf{x}$, the next reaction in

Figure 3. Algorithm for Gillespie's direct method

1: initialize: $t \leftarrow t_0$, $\mathbf{x} \leftarrow \mathbf{x}_0$
2: **repeat**
3: evaluate all propensity functions and calculate $a_0(\mathbf{x})$
4: pick 2 unit uniform random numbers n_1 and n_2
5: set $\tau \leftarrow -\ln(n_1)/a_0(\mathbf{x})$
6: set $j \leftarrow$ smallest integer satisfying $\sum_{\mu=1}^{j} a_\mu(\mathbf{x}) \geq n_2 a_0(\mathbf{x})$
7: update: $t \leftarrow t+\tau$, $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v_j}$
8: **until** simulation termination condition is met

Ω occurs in the infinitesimal time interval [t+τ,t+τ+dτ), and it is r_j. Then, it can be shown that:

$$p(\tau, j \mid \mathbf{x}, t) = a_0(\mathbf{x}) \exp(-a_0(\mathbf{x})\tau) \times \frac{a_j(\mathbf{x})}{a_0(\mathbf{x})}$$

where:

$$a_0(\mathbf{x}) \equiv \sum_{j=1}^{M} a_j(\mathbf{x}).$$

Hence, the time to the next reaction, τ, is an exponential random variable with mean $1 / a_0(\mathbf{x})$ and the index of the next reaction, *j*, is a random variable with probability $a_j(\mathbf{x}) / a_0(\mathbf{x})$. Figure 3 outlines an implementation of the SSA known as the *direct method* (Gillespie, 1976).

ALGORITHM 2.1*THE DIRECT METHOD*

Even though several streamlined implementations of the SSA have been introduced to alleviate the runtime of the stochastic simulation (e.g., Gibson and Bruck (2000); Cao et al. (2004)), the temporal behavior analyses of biochemical systems via the SSA may be very expensive. This is because the SSA can only solve the time evolution of the Markov state density function $P(\mathbf{x}, t \mid \mathbf{x_0}, t_0)$ statistically, and it may require a potentially large number of simulation runs in order to estimate the system behavior to a reasonable degree of statistical confidence. Furthermore, the SSA requires every single reaction event to be simulated one at a time, which may demand a significant amount of time, especially for realistic biochemical systems.

Simulation Abstraction

Simulation abstraction approximates the exact SSA to accelerate the simulation process while the complexity of a model is left unchanged. This approach typically involves runtime identification of reaction events that can be skipped without significant effects on the system behavior, and the usage of an approximated simulation procedure that accelerates the simulation process by sacrificing exactness.

One simulation abstraction method is Gillespie's explicit τ-leaping method (Gillespie, 2001). The basic idea of the τ-leaping method is to approximate the number of firings of each reaction in a preselected time interval τ rather than individually executing each reaction event. Thus, if τ is selected to be large enough to leap many reaction events, then the simulation process accelerates drastically. However, in order for the τ-leaping method to approximate the SSA well, the leaping time τ must be chosen so that changes in the values of the propensity functions of each reaction in the interval $[t,t+\tau]$ are kept minimal. With this condition being satisfied, the system advancement from time t to time $t+\tau$ can be well-approximated using a Poisson-distributed random variable which gives the number of r_j reaction events that fire in the time interval $[t,t+\tau)$.

The core of the τ-leaping method is the selection of τ. Hence, there have been a number of techniques introduced to improve the original τ-selection to improve the leaping method itself (e.g., Gillespie and Petzold (2003); Cao et al. (2006)). There are many variants of the τ-leaping method. Several τ-leaping methods are introduced to avoid having a molecular population go negative, which may happen in the original τ-leaping method (Tian and Burrage, 2004; Chatterjee et al., 2005; Cao et al., 2005a). The implicit τ-leaping method is introduced to better accommodate systems with stiff conditions where reactions with widely different time scales are present (Rathinam et al., 2003). The trapezoidal τ-leaping method (Cao and Petzold, 2005) is proposed to have a better accuracy and stiff stability properties than the explicit and the implicit τ-leaping methods by adapting the trapezoidal rule (Ascher and Petzold, 1998) for solving ODEs.

While the τ-leaping methods are very promising for some systems, they may not perform well for systems with fast reactions driven by species present in very small counts. This is because, in such systems, the leaping time τ which satisfies the Leaping Condition is so small that leaping many reaction events is not feasible. In such cases, the exact SSA usually performs better than the τ-leaping methods.

Another example of simulation abstraction is the *slow-scale SSA* (ssSSA) (Cao et al., 2005b). This method addresses biochemical systems with very large time scale differences (i.e., some reactions take place much less frequently than other reactions). The main idea of the ssSSA is to skip over the expensive fast reactions and simulate only the slow reactions. This is accomplished by first partitioning a system into a fast subsystem and a slow subsystem, and then by assuming that the fast subsystem rapidly reaches a well-defined stationary probability distribution with respect to the time scale of the slow subsystem. Therefore, the slow-scale propensity functions with a stationary fast subsystem can be used to predict when and which slow reaction event fires next. Although the ssSSA can efficiently approximate the stochastic simulation of some systems with large time scale differences, it has several limitations. First, it is not feasible to compute the stationary distribution of the fast subsystem for most systems, and thus it usually has to be computed approximately. Second, since the propensity functions of some reactions can change substantially during each simulation, computationally expensive partitioning of reactions and species may need to be performed frequently in such situations.

Model Abstraction

Although simulation abstraction is promising for many applications, the strict SCK model requirements may not be suitable for a large, systems-level model as the underlying system complexity does not change with this approach. Instead, *model abstraction* can be employed to transform a low-level model to a higher-level model, making computational analysis more efficient and the complexity of the system lower. While the detailed reaction-level representations of biochemical networks allow for very compre-

hensive descriptions of biological systems, such low-level models may lead to substantial computational costs and may obscure the understanding of the overall system structure and interdependency of the components. Thus, going to a higher-level representation and abstracting away dynamically insignificant reactions or species in order to reduce the complexity of the system can help make the overall systems biology analysis more efficient, as well as make crucial components and interactions of a system more intuitive. This could be accomplished through a variety of techniques depending on the structure of the system and the assumptions utilized.

An example of model abstraction is the Michaelis and Menten (MM) approximation that can reduce the dimensionality of the following enzymatic reaction scheme:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \xrightarrow{k_2} E + P \tag{4}$$

where *E*, *S*, *C*, and *P* represent an enzyme, a substrate, an enzyme-substrate complex, and a product, respectively. In the context of continuous-deterministic analysis, the theoretical basis of the MM approximation can be shown by assuming that the changes in | *C* | (i.e., the state of C) over time is minimal on the time scales of interest (Briggs and Haldane, 1925). This approximation is known as the *quasi-steady-state approximation* (QSSA), and it approximates the enzymatic reactions by the following reaction:

$$S \xrightarrow{\frac{k_2 E_{tot} |S|}{K_M + |S|}} P, \tag{5}$$

where $K_M \equiv (k_{-1} + k_2) / k_1$ which is known as the MM constant. The application of the QSSA can be justified with *singular perturbation theory* which states that the error from the QSSA estimate is small when the condition

$$\frac{E_{tot}}{| S |_0 + K_M} << 1 \tag{6}$$

is satisfied (Segel, Lee A. and Slemrod, Marshall, 1989; Keener and Sneyd, 1998). Note that $| S |_0$ represents the initial state of *S*.

Due to the substantial computational demands of the SSA, this type of approximation has recently been applied to the SCK framework. Aside from the advantage of reducing dimensionality of the system, one major advantage of the stochastic version of the MM approximation is that it can substantially reduce the simulation time by removing the fast reactions. Thus, to facilitate more efficient temporal behavior analysis, MM-type approximations have been applied to several biochemical systems (Ackers et al., 1982; Arkin et al., 1998; Wolf and Arkin, 2002). Also, the mathematical justification for the application of the *quasi-steady-state approximation* (QSSA) within the SCK framework has been investigated to establish a theoretical basis to illustrate how the QSSA can be applied to the SSA (Rao and Arkin, 2003; Samoilov, 2003).

Another example of model abstraction is a finite system state transformation of SCK models. In SCK analysis via the SSA, the temporal behavior is estimated by generating *n* sample trajectories of the system as outcomes of *n* simulation runs. Intuitively, as $n \to \infty$, this approach gives the best

estimate of the temporal behavior. Indeed, at this limit, if the system has a finite variance, the *central limit theorem* guarantees that the distribution of the *n*-sample average is asymptotically normal, and the *standard error*, S_E, which measures the difference between the estimated mean temporal behavior from the *n* Monte Carlo simulation runs and the true mean temporal behavior of the system is formulated as $S_E = \hat{\sigma} / \sqrt{n}$ where $\hat{\sigma}$ is the estimated standard deviation. This implies that, as $n \to \infty$, the numerical estimation reflects the true mean behavior. This also shows that, in order to decrease the uncertainty involved in the numerical estimation of temporal behavior *N* times, the number of simulation runs must be increased times. Instead of taking this potentially very expensive approach, the SCK model can be approximated by a finite state model and the corresponding CME can be directly solved to estimate the time evolution of the probability distribution (Kuwahara et al., 2006b; Peleš et al., 2006; Munsky and Khammash, 2006).

Although many model abstractions have long been in wide use individually, their traditionally manual transformation becomes increasingly more tedious and demanding as multiple methods are collectively applied to a particular biological system. The problem becomes even more acute as the size of the network increases, eventually rendering it intractable and potentially leading to significant errors in large model transformations. To address these issues, the remainder of this chapter describes various automated model abstraction methods.

AUTOMATED REACTION-BASED MODEL ABSTRACTION

Reaction-based abstraction methods are used to reduce a REB model's size by merging reactions, removing irrelevant reactions, etc. We have implemented several such techniques, each traversing the graph structure of the REB model and applying transformations to it when the respective conditions are satisfied. The result is a new REB model with fewer reactions and/or species. This section presents a few such methods.

Operator-Site Reduction

REB models of gene networks generally include multiple operator sites which transcription factors may occupy. It is often the case that the rates at which transcription factors bind and unbind to these operator sites are rapid with respect to the rate of *open complex formation* (i.e., initiation of transcription). It is also typically the case that the number of operator sites is much smaller than the number of *RNA polymerase* (RNAP) and transcription factor molecules. Therefore, a method similar to the QSSA and the rapid equilibrium approximation called *operator site reduction* can be used to systematically merge reactions and remove operator sites and their complexes from REB models. Note that this method may also be applicable to other molecular scaffolding systems such as those found in signal transduction networks.

The first step in this transformation is to identify operators within the REB model. This is done by assuming that an operator is a species small in number that is neither produced nor degraded. Suppose our algorithm has identified an operator *O*, and there are *N*+1 configurations in which transcription factors and RNAP can bind to it. Let O_i, K_i, and X_i with $i \in [1,N]$, be the *i*-th bound complex of the operator *O*, the equilibrium constant for forming this configuration—which is the ratio of the forward rate constant

and the backward rate constant—and the product of the states of the substrates for each component of the complex in this configuration, respectively. Let O_0 be the operator in free form (i.e., not bound to anything). Let C_i with $i \in [0,N]$ be each of the operator configurations. Then, assuming rapid equilibrium, the probability of this operator being in each configuration is:

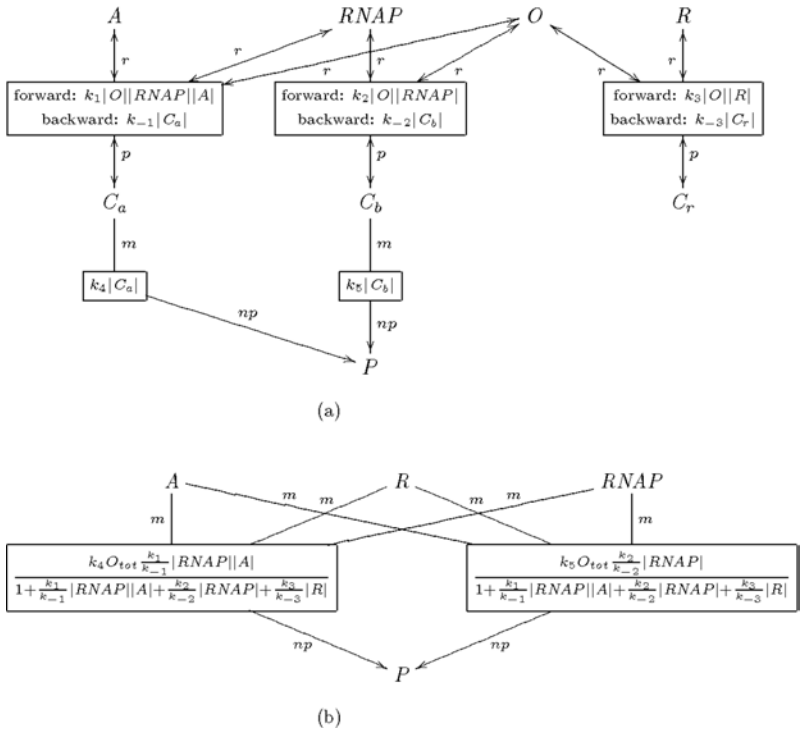
$$\Pr(C_i) = \begin{cases} \dfrac{1}{Z} & \text{if } i = 0 \\ \dfrac{K_i X_i}{Z} & \text{if } 1 \le i \le N \end{cases}$$

where $Z = 1 + \sum_{j=1}^{N} K_j X_j$. This probability is the same as the equilibrium statistical thermodynamic model when $K_i = \exp(\Delta G_i / RT)$ where ΔG_i is the relative free energies for the *i*-th configuration, *R* is the gas constant, and *T* is the absolute temperature (Ackers et al., 1982). Assuming that $O_{tot} = | O_o |_0$, then $| O_i | = \Pr(C_i) O_{tot}$ is the fraction of operators in the *i*-th configuration.

This section gives a high-level description of operator site reduction while detailed algorithms can be found in (Kuwahara et al., 2006a; Kuwahara, 2007). Operator site reduction first traverses a REB model to search for species whose initial molecule counts are small enough to be considered as potential operator sites. Each potential operator site, *O*, is checked to see if the total molecule count is conserved and if each binding reaction of *O* with transcription factors and/or RNAP, *r*, is an elementary reaction to form a complex, O_i. If these criteria are not met, then *s* is not considered further. Furthermore, if *O* or O_i is an *interesting species* (i.e., a species identified by the user as one that is being analyzed and should never be abstracted), then *O* is not considered further. The species O_i may appear as a modifier in any number of reactions that lead to synthesis of proteins. Each of these reactions, r_2, is checked that it is an elementary reaction with no reactants, only one modifier, and only one product. For each *O*, the information required to express the probabilities $\Pr(C_i)$ is stored to build the equilibrium statistical thermodynamic model. Operator site reduction then loops through the set of configurations C_i of each *O* to form an expression that is used in the denominator in each new rate law as well as forming lists of all the transcriptional regulatory proteins. Next, it considers each C_i. For each reaction r_2 in which O_i appears as a modifier, it adds all the transcription factors as modifiers and creates a new rate law for r_2. Finally, the reduction removes all the binding/unbinding reactions of *O*, all the complex species O_i, and the operator site *O* from the model.

As an example, Figure 4(a) shows the graphical representation of a detailed REB model which describes transcriptional gene regulation to produce protein *P* based on the configurations of operator site *O* bindings. In this graphical representation, a reaction that is connected to a species with a double arrow is a shorthand to show a *reversible reaction* (i.e., a pair of two reactions with reactants and products swapped). Species connected to a reaction with letters, *r*, *p*, and *m* are a reactant, a product, and a modifier for that reaction, respectively. The math expression inside a reaction node is the kinetic rate function of that reaction. In Figure 4(a), the top three reversible reactions involve the binding of *RNAP*, an activator *A*, and repressor *R* to *O* while the bottom two irreversible reactions result in the production of *n* molecules of the protein *P*. In this example, there are 4 configurations of the operator, namely, *O*, C_a, C_b, and C_r. This network has eight species and eight irreversible reactions. Assuming that the operator-binding and unbinding rates are much faster than those of open complex formation, our method can apply operator site reduction. Figure 4(b) is the result of applying this abstraction method to Figure 4(a). The result has only three species and two reactions. The transformed model represents the probability of *O* being

Figure 4. Operator site reduction: (a) original model and (b) abstracted model

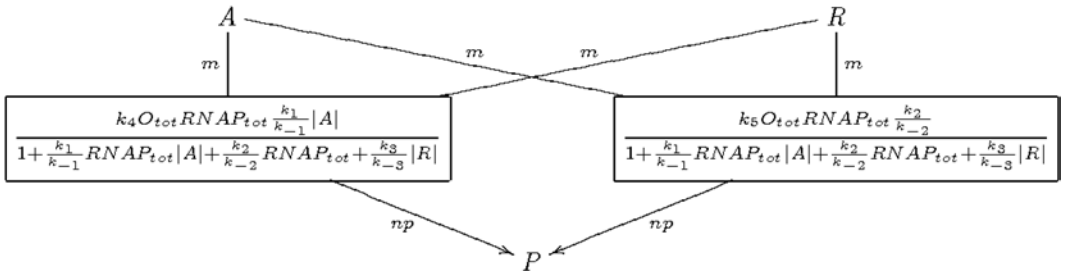


in a configuration that results in production of *P* instead of modeling every binding and unbinding of transcription factors and *RNAP* to the promoter precisely.

Modifier Constant Propagation

In order to increase the understandability of a REB model as well as the efficiency of its temporal behavior analysis, it is essential to remove all *unimportant* species that do not contribute to the dynamics of a system. In particular, it is useful to systematically inspect and remove species whose states are statically known to stay unchanged in simulation which is accomplished using an abstraction method

Figure 5. A REB model after applying modifier constant propagation to a REB model shown in Figure 4(b)



called *modifier constant propagation*. Modifier constant propagation traverses a REB model, and finds a species *s* which is only used as a modifier. It then substitutes a constant $|S|_0$ for $|S|$ in the kinetic law expressions of the reactions that use *s* as a modifier. Therefore, since $|S|$ is no longer used to influence any kinetic laws, species *s* is safely removed from a REB model by this method.

For example, as illustrated in the REB model in Figure 4(b), after applying operator site reduction, it is often the case that *RNAP* is only used as a modifier. Thus, by applying modifier constant propagation, can be replaced with a constant $RNAP_{tot}$ where $RNAP_{tot} = | RNAP |_0$. Therefore, as shown in Figure 5, the REB model in Figure 4(b) can be reduced to three species and two reactions as a result of modifier constant propagation.

Similar Reaction Combination

A REB model may contain multiple reactions whose structures are very similar. Thus, combining such reactions using another abstraction method called *similar reaction combination* can improve the complexity of a REB model by reducing the number of reactions in a REB model. It can also result in a reduction of the computational costs for evaluating kinetic laws by reducing redundant kinetic law expressions. In the context of gene regulatory networks, an abstracted REB model of transcriptional gene regulation often has protein synthesis mechanisms at a basal rate and enhanced or reduced rates due to transcription factors binding to operator sites. These mechanisms can be represented in structurally similar reactions whose kinetic laws typically contain redundant expressions. Thus, with this method, such protein synthesis mechanisms can be combined into one reaction with a computationally much less expensive kinetic law expression.

Similar reaction combination transforms a REB model by first searching for structurally similar reactions and replaces them with one reaction. Here, reactions r_1 and r_2 are defined to be structurally similar if reactions r_1 and r_2 have the same reactants, products, and modifiers with the same stoichiometries. An implication of this condition is that firings of both reactions r_1 and reaction r_2 are guaranteed to result in the same state transition of a REB model. Thus, these reactions can be combined to introduce a new reaction r_c such that:

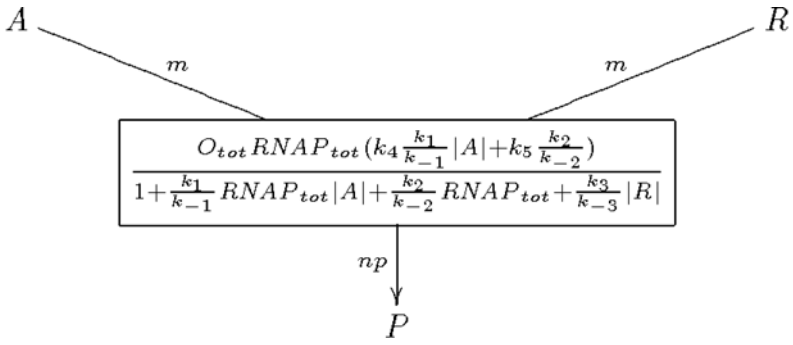
$$\mathbf{K}(r_c) = \mathbf{K}(r_1) + \mathbf{K}(r_2) \tag{7}$$

where $\mathbf{K}(r)$ represents the kinetic law for reaction *r*. The ODE model of the combined reactions is identical to the original one. Hence, the similar reaction combination method can be used without making any approximation in the continuous-deterministic analysis case. In the case of SCK analysis via the SSA, suppose reactions r_1 and r_2 are structurally similar. Then, from Condition 7, the probability that either reaction r_1 or reaction r_2 is chosen to be the next reaction to fire becomes:

$$\mathrm{Prob}(r_1 \text{ or } r_2) = \frac{\mathbf{K}(r_1)}{\sum_{r\in\mathbf{R}} \mathbf{K}(r)} + \frac{\mathbf{K}(r_2)}{\sum_{r\in\mathbf{R}} \mathbf{K}(r)} = \frac{\mathbf{K}(r_c)}{\sum_{r\in\mathbf{R}} \mathbf{K}(r)}. \tag{8}$$

Thus, the probability of firing an event of the newly introduced reaction r_c is identical to the probability of firing an event of either reaction r_1 or r_2. Similarly, from Condition 7, the computation of the next reaction time τ in the direct method of the SSA does not change before and after the reaction

Figure 6. A REB model after applying similar reaction combination to a REB model shown in Figure 5



combination as the sum of the propensity functions does not change. Furthermore, the state transitions via the combined reaction are the same as that of reactions r_1 and r_2. Therefore, this method itself does not make any approximation for the SCK analysis as well.

To illustrate an application of similar reaction combination, Figure 6 shows the REB model that is abstracted from the REB model in Figure 5 by using similar reaction combination. In this reduced REB model, the two reactions to produce *n* molecules of protein *P* are combined into one reaction. Since the kinetic laws of the two reactions have the same denominator, the combined reaction is able to simplify its kinetic law, making its evaluation faster than that of the two original kinetic laws.

Other Reaction-Based Model Abstractions

Besides the three REB abstraction methods described in this chapter, we have implemented several additional ones listed below (Kuwahara et al., 2006a; Kuwahara and Myers, 2007; Kuwahara, 2007):

- *Irrelevant node elimination* removes species and reactions that do not significantly influence the species of interest.
- *Production-passage-time approximation* approximates the production time in an enzymatic reaction scheme to remove the expensive complex-dissociation reaction (Kuwahara and Myers, 2007).
- *Quasi-steady-state approximation* merges reactions and removes intermediate species in an enzymatic reaction scheme when the quasi-steady-state assumption holds.
- *Rapid equilibrium approximation* also merges reactions and removes intermediate species in an enzymatic reaction scheme when it has large time scale difference characteristics.
- *Dimerization reaction reduction* removes dimerization reactions and expresses the number of monomers and dimers in terms of its total number of molecules.
- *Stoichiometry amplification* amplifies stoichiometries and reduces the values of propensity functions making the system and time advancement per reaction larger.
- *Reaction splitizations* split reactions so that each reaction changes the state of only one species.

AUTOMATED STATE-BASED MODEL ABSTRACTION

The state-based model abstractions convert the REB model into a FSS model. The FSS model is a state-based, continuous-time discrete-event system where the stochastic state transition of each species is restricted to a finite state space unlike most state transitions in a REB model. By making each species' state space finite, the overall system space of the FSS model becomes also finite. Thus, the state space of a FSS model can be explicitly specified. The FSS model compactly represents a time-homogeneous, discrete-state, Markov process in a finite state space whereby state transitions are decided based on the information on the current state. Therefore, while a system described using the FSS model can be analyzed via stochastic simulation methods such as the SSA, it can also be analyzed using a Markov chain analysis method (Stewart, 1994)—albeit possibly requiring a substantial amount of memory to generate all the underlying system states—to directly obtain the solution of an abstracted CME. The FSS model is formally defined as follows.

Definition 4.1 (FSS model) A FSS model is specified with $\left\langle \mathbf{Z}, \mathbf{z}_0, \mathbf{z}_{\max}, \mathbf{C} \right\rangle$ where $\mathbf{Z} \equiv (Z_1, \ldots, Z_n)$ is a vector of non-negative integer random variables, $\mathbf{z}_0$ is the vector containing the values of $\mathbf{Z}$ at time 0, $\mathbf{z}_{\max}$ is the vector whose i-th element, $z^i_{\max}$, specifies the maximum value that random variable Z_i can take, and $\mathbf{C} \equiv \left\{c_1, \ldots, c_m\right\}$ is the set of guarded commands that change the values of the random variables. The system state space of $\mathbf{Z}$, $\mathbf{\Sigma}_\mathbf{Z}$, is specified as

$$\mathbf{\Sigma}_\mathbf{Z} = \left\{ \mathbf{z} \mid \forall i.\ z_i \in 0, z^i_{\max}] \right\}.$$

$\mathbf{Z}(t)$ specifies the system state at time t. Thus, for each Z_i, the probability that $Z_i(t) > z^i_{\max}$ or $Z_i(t) < 0$ is zero for any $t \geq 0$. When the system is in state $\mathbf{z}$, each guarded command, c_j, has a form:

$$G_j(\mathbf{z}) \xrightarrow{q_j} \mathbf{Z} = \mathbf{z} + \mathbf{u}_\mathbf{j}$$

where the function $G_j(\mathbf{z}) : \left\{0, \ldots, z^i_{\max}\right\}^n \mapsto \left\{0,1\right\}$ is the guard for c_j when the system state is $\mathbf{z}$, q_j is the transition rate for c_j, and $\mathbf{u}_\mathbf{j}$ is an n-dimensional vector whose i-th element has the value added to Z_i as a result of c_j.

Let [[*bool−exp*]] be an operator that takes a Boolean expression, *bool−exp*, and evaluates to 1 if *bool−exp* is true and 0 otherwise. Then, the guard, $G_j(\mathbf{z})$, of each guarded command, c_j, has the form:

$$G_j(\mathbf{z}) = \coprod_{i \in \mathbf{N}_\mathbf{j}} [[z_i = v^i_j]] \tag{9}$$

where the expression $[[z_i = v^i_j]]$ results in 1 if the current state of Z_i is equal to the value specified by the constant v^i_j, otherwise results in 0, and $\mathbf{N}_\mathbf{j}$ is a subset of $[1,n]$. Each guarded command, c_j, is required to change the state of $\mathbf{Z}$. Thus, each $\mathbf{N}_\mathbf{j}$ must satisfy the condition $\left|\mathbf{N}_\mathbf{j}\right| > 0 \wedge \left|\mathbf{N}_\mathbf{j}\right| \leq n$. If the system state is $\mathbf{z}$ at time t (i.e., $\mathbf{Z}(t) = \mathbf{z}$), can be executed if its guard is satisfied (i.e., $G_j(\mathbf{z}) = 1$). The result of executing the guarded command in time step τ is that a new state is reached in which $\mathbf{Z}_\mathbf{i}(t + \tau) = \mathbf{z} + \mathbf{u}_\mathbf{j}$. Note that $G_j(\mathbf{z})$ can be efficiently encoded in the state graph of a FSS model by using the connection from state $\mathbf{z}$ to state $\mathbf{z} + \mathbf{u}_\mathbf{j}$ as an indicator so that $G_j(\mathbf{z})$ is evaluated to 1 if there is a transition edge from state $\mathbf{z}$ to state $\mathbf{z} + \mathbf{u}_\mathbf{j}$, otherwise to 0.

From the definition of the FSS model, the probability that, given the system is in state $\mathbf{z}$, c_j is executed and $\mathbf{Z}$ moves to state $\mathbf{z} + \mathbf{u}_j$ within the next infinitesimal time step *dt* is: $P(c_j, dt \mid \mathbf{z}) = G_j(\mathbf{z}) q_j dt$. By taking the limit: $dt \to 0$, the following abstracted CME represented by a FSS model can be obtained:

$$\frac{\partial P(\mathbf{z}, t \mid \mathbf{z}_0)}{\partial t} = \sum_{j=1}^{m} [G_j(\mathbf{z} - \mathbf{u}_j) q_j P(\mathbf{z} - \mathbf{u}_j, t \mid \mathbf{z}_0) - G_j(\mathbf{z}) q_j P(\mathbf{z}, t \mid \mathbf{z}_0)]. \quad (10)$$

4.1 Finite State System Model Transformation

In order to describe the transformation from a REB model, M_R, to a FSS model, M_F, let us suppose that M_R has $\mathbf{S} \equiv \{s_1, \ldots, s_n\}$ where the state of each species s_i can be changed by some reaction, $\mathbf{R} \equiv \{r_1, \ldots, r_{m'}\}$ where each reaction r_j can change the state of some species. Then, M_F has $\mathbf{Z} \equiv \{Z_1, \ldots, Z_n\}$ where each Z_i specifies $|s_i|$, and each z_0^i in $\mathbf{z}_0$ is $|s_i|_0$. Each $z_{\max}^i$ in $\mathbf{z}_{\max}$ is set by the user to specify the upper limit molecular count of s_i.

For the generation of the guarded commands, $\mathbf{C} \equiv \{c_1, \ldots, c_m\}$, each reaction r_j first constructs a set of indices, $\mathbf{I}_j$, that contains all the indices of the species that participate in reaction r_j. Here, let $\hat{\mathbf{Z}}_j$ be the set of random variables $\{Z_{i'} \mid i' \in \mathbf{I}_j\}$, and $\mathbf{\Sigma}_{\hat{\mathbf{Z}}_j}$ be a subset of $\mathbf{\Sigma}_\mathbf{Z}$ which has every state $\mathbf{z}$ in which, using the value of each z_i for $|s_i|$, a reaction r_j event can fire with a non-zero transition rate to move to a state where the new value of each Z_i is at most $z_{\max}^i$ and at least 0, provided that the states of species that do not participate in reaction r_j are fixed to be 0 (i.e., $z_i = 0$ if $i \in \mathbf{I}_j$). Then, each reaction r_j generates guarded commands using the information on each state in $\mathbf{\Sigma}_{\hat{\mathbf{Z}}_j}$, resulting in as many as $|\mathbf{\Sigma}_{\hat{\mathbf{Z}}_j}|$ guarded commands. Each guarded command, c_μ, from state $\mathbf{z}_\mu \in \mathbf{\Sigma}_{\hat{\mathbf{Z}}_j}$ is used for the transition event of reaction r_j in states where only the value of $Z_j \in \hat{\mathbf{Z}}_j$ is constrained by the *i*'-th element of $\mathbf{z}_\mu$, $z_\mu^{i'}$. The guard, $G_\mu(\mathbf{z})$, checks if the condition to enable the transition event, c_μ, is satisfied in state $\mathbf{z}$. This condition can only be satisfied when

$$\forall i' \in \mathbf{I}_j.\ z_{i'} = z_\mu^{i'} \quad (11)$$

is true. To generate the form of Equation 9, thus, $\mathbf{N}_\mu$ is set so that $\mathbf{N}_\mu$ is equal to $\mathbf{I}_j$, and each v_μ^i is the constant whose value is specified by z_μ^i. The transition rate of c_μ, q_μ, is computed by evaluating $\mathbf{K}(r_j)$ using the value of $\mathbf{z}_\mu$. The increment vector, $\mathbf{u}_\mu$, specifies the increment of the system state as a result of the firing of one event. Thus is generated so that the *i*-th element of $\mathbf{u}_\mu$ is specified as net amount of produced by reaction r_j.

4.2 N-ary Transformation

While the FSS model transformation method described in the previous section provides a means to analyze the time evolution of biochemical systems by directly solving the CMEs, this method is proven to be inefficient for systems with very large system state space. Even for a system of 10 species where each has an upper limit molecular count of 99, the FSS model transformation can generate up to 10^{20}

states. Constructing such a state graph for temporal behavior analysis is infeasible for most computers. Thus, the FSS model transformation method should not be used in such cases. To more aggressively reduce the state space of a FSS model, this section develops another transformation method called *n-ary transformation*. The n-ary transformation transforms a REB model to a reduced FSS model called the *stochastic asynchronous circuit* (SAC) model. A SAC model describes the state of each species by n-ary or Boolean levels instead of molecular counts, resulting in further reduction of states per species. Thus, it can further improve the analysis time. For example, suppose a system has 10 species, each of whose states can be qualitatively described as *low*, *medium*, and *high*. Then, with the n-ary transformation, a SAC model with at most 10^3 states can be generated. Therefore, this model can be efficiently analyzed, for example, using Markov chain analysis methods within the asynchronous circuit analysis tool ATACS (Myers et al., 2001).

Aside from the conditions required for the FSS model transformation, the n-ary transformation requires the REB model to satisfy the property that all reactions should have either one reactant *or* one product, but not both. Thus, each guarded command in a SAC model, c_j, comes with the following restriction on the increment vector, $\mathbf{u_j}$:

$$\exists i.\ (u_j^i > 0) \wedge (\forall k \neq i.\ u_j^k = 0). \tag{12}$$

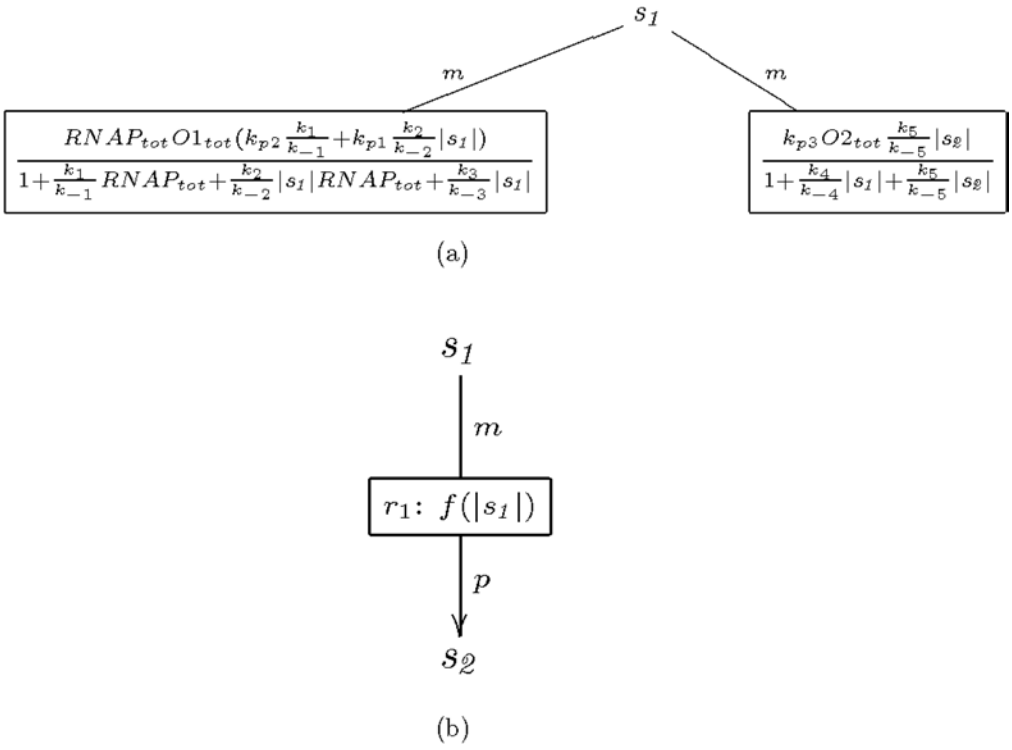
Thus, the SAC model is a subset of the FSS model whereby each guarded command changes the value of exactly one random variable as a result of its execution. This is often the case after applying the REB abstractions described earlier. If this condition does not hold, however, it can be made to hold using reaction splitizations.

The n-ary transformation begins by identifying the states of each species. Let $\mathbf{A_i} \equiv \left\{A_i^0, A_i^1, ..., A_i^{N_i}\right\}$ be a set with N_i elements that partitions the states of species s_i such that $\forall j.\ A_i^j = [\theta_i^j, \theta_i^{j+1})$ where $\theta_i^0 = 0$, and $\theta_i^{N_i+1} = \infty$. We call $A_i^0, ..., A_i^{N_i}$ *critical intervals* and $\theta_i^0, ..., \theta_i^{N_i}$ *critical levels* of species s_i. Depending on the nature of the application, the critical levels can be either specified by the user and taken to be model inputs—such as might be the case when our system is utilized by an expert already familiar with the *in situ* behavior of the underlying regulatory network—or estimated automatically from the kinetic rate laws. The SAC model treats each A_i^j as one state. Thus, N_i describes the highest state for species s_i in a SAC model, and thus, in the FSS model notation, N_i is in fact $z_{\max}^i$ in that, for all $t \geq 0$, $Z_i(t) \geq 0 \wedge Z_i(t) \leq N_i$. The initial state of Z_i, z_0^i, is determined by examining each critical interval, A_i^j for the condition: $\left|s_i\right|_0 \in A_i^j$. Then, z_0^i is set to the index of the critical interval that satisfies this condition.

In $\mathbf{A_i}$, if $\theta_i^{j+1} - \theta_i^j >> 1$ for some j in $\left[1, N_i\right]$, then our method can collapse many states in A_i^j into one state for species s_i, resulting in significant improvement in analysis time. On the other hand, if $\forall j \in \left[1, N_i\right].\ \theta_i^{j+1} - \theta_i^j = 1$ for all i, then each state of the SAC model describes a molecular count, resulting in the same precision in the state space of species s_i as the FSS model.

In order to identify the critical levels of species s_i, our method first automatically finds all reactions with kinetic rate laws that include a denominator term of the form $K\left|s_i\right|^n$. For each such reaction, one critical level of s_i is generated with the form $\sqrt[n]{a\ /\ (K - aK)}$ where a is an amplifier in the range [0.5,1.0) selected by the user. Figure 7(a) shows two reactions that have kinetic rate laws containing $\left|s_1\right|$ terms.

Figure 7. (a) Critical level identification. (b) Production of s_2 with activator s_1. $f(|s_1|) > 0$ if $|s_1| \geq 0$.



Assuming that amplifier *a* equals 0.5, these two reactions imply the following four critical levels:

$$0, \frac{k_{-4}}{k_4}, \frac{k_{-2}}{k_2 \cdot RNAP_{tot}}, \text{ and } \frac{k_{-3}}{k_3} \tag{13}$$

These levels come from the fact that θ_0 is by definition 0, the denominator of the left reaction rate law in Figure 7(a) has the term $k_4 / k_{-4} |s_1|$, and the denominator of the right reaction rate law has two terms of this form, $k_2 / k_{-2} |s_1| RNAP_{tot}$ and $k_3 / k_{-3} |s_1|$.

After the critical levels of each species are identified and in turn $\mathbf{z_0}$ and $\mathbf{z_{max}}$ are all determined, the guard, $G_\mu(\mathbf{z})$, for c_μ is generated for each reaction in a similar way as the FSS model transformation. Suppose species s_1 is an activator in reaction r_1 for the production of s_2 as shown in Figure 7(b) where its kinetic law, $f(|s_1|)$, is always greater than 0 if $|s_1| \geq 0$. Also, suppose that three critical levels are used for both species s_1 and s_2, that is, the critical levels of s_1 and s_2 are $(0, \theta_1^1, \theta_1^2)$, and $(0, \theta_2^1, \theta_2^2)$, respectively.

In the n-ary transformation, definition of $\mathbf{\Sigma}_{\hat{\mathbf{z}}_j}$ is slightly different from the FSS model transformation. Since r_1 is a production reaction for species s_2 where $\mathbf{\Sigma}_{\hat{\mathbf{z}}_j}$ is obtained as:

$$\mathbf{\Sigma}_{\hat{\mathbf{z}}_j} = \left\{(z_1, z_2, ..., z_n) \mid z_1 \in [0,2] \text{ and } z_2 \in 0,1]\right\}, \tag{14}$$

the legal transitions that Z_2 can take are only two: $0 \rightarrow 1$ and $1 \rightarrow 2$ for each possible state of Z_1. Therefore, the guarded commands for r_1 are below:

$$\begin{aligned}
&\left[\left[z_2 = 0\right]\right] \times \left[\left[z_1 = 0\right]\right] \xrightarrow{q_1} \mathbf{Z} = \mathbf{z} + \mathbf{u} \\
&\left[\left[z_2 = 0\right]\right] \times \left[\left[z_1 = 1\right]\right] \xrightarrow{q_2} \mathbf{Z} = \mathbf{z} + \mathbf{u} \\
&\left[\left[z_2 = 0\right]\right] \times \left[\left[z_1 = 2\right]\right] \xrightarrow{q_3} \mathbf{Z} = \mathbf{z} + \mathbf{u} \\
&\left[\left[z_2 = 1\right]\right] \times \left[\left[z_1 = 0\right]\right] \xrightarrow{q_4} \mathbf{Z} = \mathbf{z} + \mathbf{u} \\
&\left[\left[z_2 = 1\right]\right] \times \left[\left[z_1 = 1\right]\right] \xrightarrow{q_5} \mathbf{Z} = \mathbf{z} + \mathbf{u} \\
&\left[\left[z_2 = 1\right]\right] \times \left[\left[z_1 = 2\right]\right] \xrightarrow{q_6} \mathbf{Z} = \mathbf{z} + \mathbf{u}
\end{aligned}$$

where $\mathbf{u} \equiv (u_1,...,u_n)$ has $u_2 = 1$ and $\forall i \neq 2.\ u_i = 0$.

The final step to generate a SAC model is to assign a transition rate, q_i, to each guarded command. For simplicity, N_i Boolean variables $B_i^1,...,B_i^{N_i}$ are introduced for the generation of the rate to change the state of Z_i. The relationship between Z_i and $B_i^1,...,B_i^{N_i}$ is:

$$Z_i(t) = z \text{ iff } (\forall \text{j} \in [1,\text{z}].\ B_i^j(t) = 1) \wedge (\forall j \in [z+1, N_i].\ B_i^j(t) = 0). \tag{15}$$

Thus, the time evolution of $\left|s_i\right|$ can be approximated using $B_i^1,...,B_i^{N_i}$ as

$$\left|s_i\right|(t) \approx (\theta_i^{N_i} - \theta_i^{N_{i-1}})B_i^{N_i}(t) + \cdots + (\theta_i^2 - \theta_i^1)B_i^2(t) + (\theta_i^1 - \theta_i^0)B_i^1(t). \tag{16}$$

Taking the derivative of the mean of $\left|s_i\right|(t)$ with respect to the mean of $B_i^j(t)$ results in:

$$\frac{\partial \left\langle \left|s_i\right|(t) \right\rangle}{\partial \left\langle B_i^j(t) \right\rangle} \approx \theta_i^j - \theta_i^{j-1}. \tag{17}$$

Using this approximation, the time derivative of $\left\langle B_i^j(t) \right\rangle$ is:

$$\frac{d\left\langle B_i^j(t) \right\rangle}{dt} = \frac{\partial \left\langle B_i^j(t) \right\rangle}{\partial \left\langle \left|s_i\right|(t) \right\rangle} \frac{d\left\langle \left|s_i\right|(t) \right\rangle}{dt} \approx \frac{1}{\theta_i^j - \theta_i^{j-1}} \frac{d\left\langle \left|s_i\right|(t) \right\rangle}{dt}. \tag{18}$$

Notice $\left\langle B_i^j(t) \right\rangle$ is a continuous variable in the range [0,1]. By letting $\left\langle B_i^j(t) \right\rangle$ be the probability that $B_i^j = 1$ at *t*, our method finds the transition rate functions for B_i^j to move from 0 to 1 and from 1 to 0 from the rate laws of reactions that change the value of $\left|s_i\right|$. The transition rate function of a guarded command changing the value of B_i^j, which is generated from reaction *r*, is:

$$f = \frac{E \cdot \mathbf{K}(r)}{\theta_i^j - \theta_i^{j-1}} \tag{19}$$

where *E* is the stoichiometry of species *i* in reaction *r*.

Finally, our method must evaluate the transition rate functions with appropriate values to generate the transition rates. Suppose reaction r_j uses $\left|s_i\right|$ in its kinetic law. Then, to generate the transition rate for the guarded command when $Z_i = z$, our method uses θ_i^z as the value of $\left|s_i\right|$ to evaluate $\mathbf{K}(r_j)$. For

example, the transition rates of the guarded commands in Figure 7(b) are derived from $\mathbf{K}(r_1)$. Since the derived transition rate function is $f(|s_1|) / (\theta_2^{\mu} - \theta_2^{\mu-1})$ for $\mu \in [1,2]$, the transition rates for the guarded commands for reaction r_1 are:

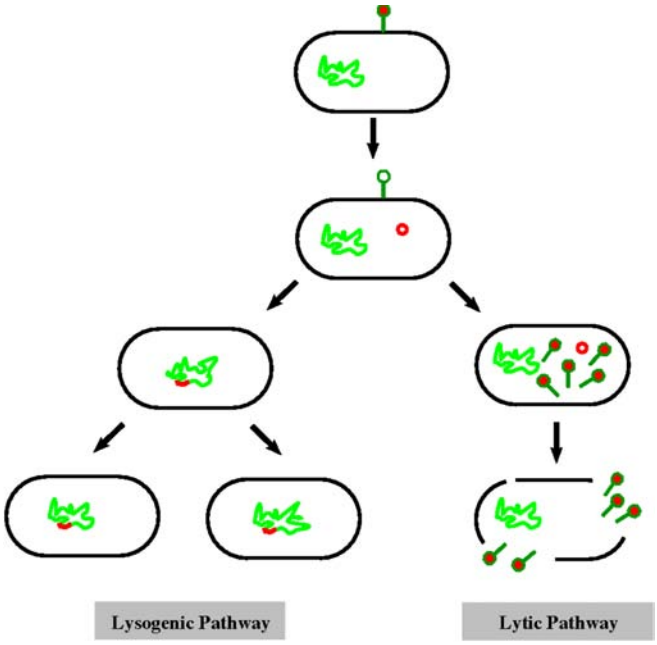
$$
\begin{aligned}
q_1 &= f(0) / \theta_2^1, \\
q_2 &= f(\theta_1^1) / \theta_2^1, \\
q_3 &= f(\theta_1^2) / \theta_2^1, \\
q_4 &= f(0) / (\theta_2^2 - \theta_2^1), \\
q_5 &= f(\theta_1^1) / (\theta_2^2 - \theta_2^1), \\
q_6 &= f(\theta_1^2) / (\theta_2^2 - \theta_2^1).
\end{aligned}
$$

CASE STUDY: PHAGE λ

Both the REB and FSS abstraction methods coupled with temporal behavior analysis methods are implemented in our automated modeling and analysis tool called REB2SAC (Kuwahara et al., 2005, 2006a) which is integrated within our iBioSim tool (available from http://www.async.ece.utah.edu/iBioSim) that provides a user-friendly graphical user interface. This section presents the application of our tool to the analysis of the phage λ-infected *E. coli* lysis/lysogeny decision switch, examining the changes in the probability of the pathway taken based on various environmental conditions. Phage λ is a virus that infects *E. coli* cells. It has two strategies to replicate itself as shown in Figure 8. One is called *lysis* where the phage creates copies of itself inside the cell and bursts the cell to escape and infect other cells. The other one is a more passive approach called *lysogeny* where the phage integrates its DNA into the host chromosome and replicates its DNA through cell division.

The genetic circuit controlling the phage λ lysis/lysogeny decision is shown in Figure 9. The key proteins involved in the phage λ lysis/lysogeny developmental decision are *CI*, *Cro*, *N*, *CII*, and *CIII*. The lysis/lysogeny decision is a race condition between *CI* and *Cro*. A high concentration of *CI* leads to the lysogenic pathway, while a high concentration of *Cro* leads to the lytic pathway. The core component of the genetic circuit is the three operator sites called the λ switch to which *CI* and *Cro* dimers can competitively bind to influence the activities of the promoters P_{RM} and P_R (Ptashne, 1992). Binding of the *CI* dimer to the λ switch in a wild type setting represses the transcription of the *cro* gene by preventing *RNAP* from binding to P_R. When the concentration of *CI* dimer is low, the operator sites in the λ switch tend to be empty, and the expression of gene *cI* from P_{RM} only occurs at a low basal rate. When the concentration of *CI* dimer is medium, it tends to occupy two operator sites in the λ switch, leading to an increased activated expression of *cI* from P_{RM}. However, when the concentration of *CI* dimer is very high, it tends to occupy all three operator sites in the λ switch, which represses the expression of *cI* from P_{RM} by preventing *RNAP* from binding to P_{RM}. Binding of *Cro* dimer to the λ switch in the wild type represses the transcription of *cI* by preventing *RNAP* from binding to P_{RM}. When the concentration of *Cro* dimer is low, the operator sites in the λ switch tend to be empty, but since *cro* expression from P_R does not need to be activated, its production proceeds at a high rate. As the concentration of *Cro* dimer increases, it tends to occupy the operator sites closest to P_{RM} shutting off *CI* production. Very high concentration of *Cro* though turns off production of both *CI* and *Cro*.

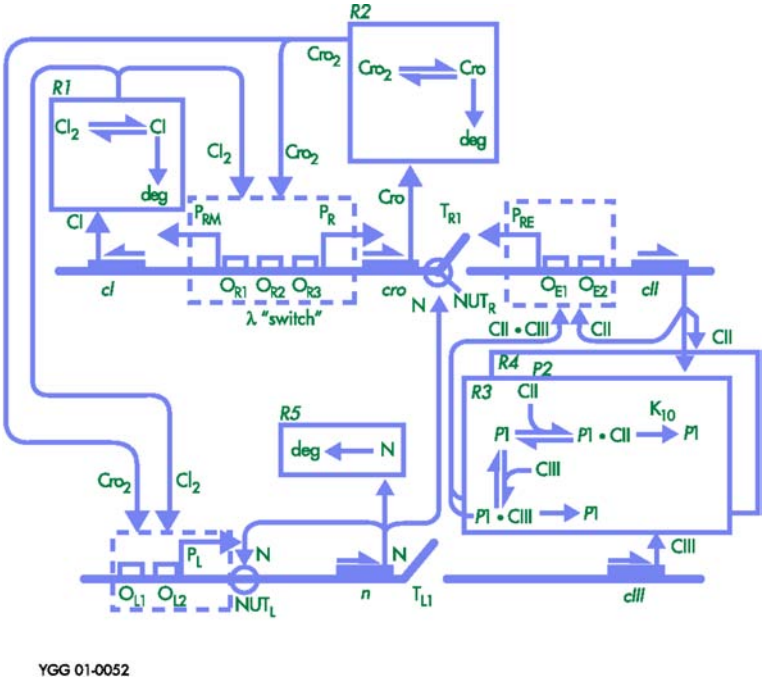
Figure 8. Phage λ lysis/lysogeny developmental pathway. Phage λ has two pathways to multiply itself called the lytic pathway and the lysogenic pathway. In the lytic pathway, the phage first creates proteins needed for formation of new viruses. It then replicates its DNA to create new viruses inside the cell. These viruses burst the cell to escape to infect other cells. In the lysogenic pathway, the phage integrates its DNA into the host chromosome. It then replicates its DNA passively via cell division.



Immediately after infection, there are no *CI* and *Cro* molecules in the cell (Arkin et al., 1998). In this condition, while *CI* can be synthesized from two promoters, P_{RM}, and P_{RE}, the synthesis of *Cro* is higher than that of *CI* since the basal transcription rate of *cro* is higher. Thus, the favored outcome of the lysis/lysogeny decision is lysis at the early stage of the decision. In order for the phage to take the lysogenic pathway, the enhanced transcription of *cI* from the P_{RE} promoter which is activated by the presence of *CII* is required. For this to happen, the antiterminator, *N*, needs to be synthesized at the early stage of the decision so that it can help *RNAP* go through the termination sites, T_{L1} and T_{R1} to facilitate synthesis of *CII* and *CIII* at the early stage of the decision. Since *CIII* can prevent *CII* from degrading, a high concentration of *CIII* can lead to a high concentration of *CII*.

We have constructed a REB model for the phage λ decision circuit system which is described in Kuwahara et al. (2006a). Our initial REB model includes 55 species and 69 reactions, and the set of interesting species, $\mathbf{S}_i$, includes *CI* and *Cro*. This model is then automatically abstracted using REB-2SAC. The abstraction engine in REB2SAC is configured for the phage λ decision circuit model so that it collectively applies REB abstraction methods as shown in Figure 10.

Figure 9. Phage λ decision circuit. The key proteins involved in the phage λ lysis/lysogeny developmental decision are CI, Cro, N, CII, and CIII. The lysis/lysogeny decision is a race condition between the states of CI and Cro. A high concentration of CI leads to the lysogenic pathway, while a high concentration of Cro leads to the lytic pathway. In order for the phage to take the lysogenic pathway, antiterminator, N, needs to be synthesized at the early stage of the cell cycle. This antiterminator can help RNAP go through the termination sites, T_{L1} and T_{R1}, facilitating transcriptions of genes cII and cIII. CIII can prevent CII from degrading by binding to proteases P1 and P2 (Arkin et al., 1998), and CII activates the transcription of cI from the P_{RE} promoter. Further description of the genetic circuit can be found in Arkin et al. (1998) (image courtesy of U.S. Department of Energy Genomics: GTL Program http://genomicsgtl.energy.gov).



ALGORITHM 5.1*MODEL ABSTRACTIONENGINE(MODEL M)*

The seven abstraction methods are applied iteratively until there is no change in the model. Irrelevant node elimination and modifier constant propagation are applied first to reduce the complexity of the model without compromising accuracy. The rapid equilibrium approximation is applied before the standard quasi-steady-state approximation so that, whenever the model contains patterns that match the conditions for both methods, the former has precedence in order to reduce the complexity of the reaction rate laws. The similar reaction combination is applied right after the operator site reduction to immediately combine the structurally similar reactions that are often generated by operator site reduction. The dimerization reduction is placed after operator site reduction since an operator site with a dimer molecule as a transcription factor cannot be reduced otherwise. After collectively applying the REB abstraction methods, the REB model is reduced to only 5 species and 11 irreversible reactions as shown graphically in Figure 11. This figure shows the biological gene-regulatory network of the phage λ lysis/lysogeny

Figure 10. Top level abstraction algorithm of the phage λ decision circuit model

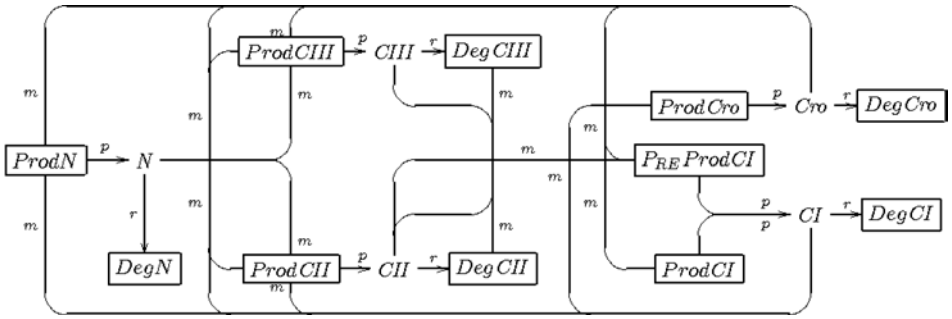
```
1: repeat
2:    M' ← M
3:    M ← Irrelevant_Node_Elimination(M)
4:    M ← Modifier_Constant_Propagation(M)
5:    M ← Rapid_Equilibrium_Approximation(M)
6:    M ← Standard_QSSA(M)
7:    M ← Operator_Site_Reduction(M)
8:    M ← Similar_Reaction_Combination(M)
9:    M ← Dimerization_Reduction(M)
10: until M' = M
11: return M
```

decision circuit, and it is quite similar to the high-level hand-generated diagram in Figure 9. The structure of this graph, however, is automatically generated using abstractions from the low level model.

The goal of our analysis using this computational model is to determine the probability that the lysogenic pathway is chosen under various conditions. For example, it has been shown experimentally that the probability of lysogeny increases as the *multiplicity of infection* (MOI)—the number of phages simultaneously infecting the same cell—increases (Kourilsky, 1973). Thus, our analysis first predicts the effects of MOI on the probability of lysogeny. For this analysis, both the original model and the abstracted one are simulated for 10,000 runs using the same simulator, an optimized implementation of SSA within REB2SAC, on a 3GHz Pentium4 with 1GB of memory to have a reasonable statistical confidence as well as to measure the speedup gained via abstractions. Each simulation is run for up to one cell cycle while tracking the number of molecules of *CI* and *Cro*. If the number of *CI* molecules exceeds 328 (i.e., 145 *CI* dimers) before the number of *Cro* molecules exceeds 133 (i.e., 55 *Cro* dimers), then the simulation run is said to result in lysogeny (Arkin et al., 1998). The simulations are run for MOIs ranging from 1 to 50. While the simulation of the original REB model takes 56.5 hours, the abstracted model takes only 9.8 hours, which is a speedup of more than 5.7 times. Figure 12(a) shows the probability of lysogeny for MOIs from 0 to 10 for both the original REB model and the abstracted one. The results are nearly the same, yet with a substantial acceleration in runtime.

The n-ary transformation is able to automatically convert our reduced REB model for the phage λ

Figure 11. Structure of the abstracted model of the phage λ developmental decision gene-regulatory pathway



decision circuit into a reduced FSS (SAC) model. However, since the species *CI* and *Cro* influence many reactions, our automated analysis finds that 10 critical levels are needed for species *CI*, and 10 are needed for species *Cro*. This is too many critical levels for the Markov chain analyzer within ATACS (Myers et al., 2001). Fortunately, many of these critical levels are very close together and can be combined with little loss in accuracy. Therefore, while we decided to use nine levels for species *CI* and four levels for *CII*, we used only two levels for each of the species *Cro*, *N*, and *CIII*.

We analyzed the SAC model using Markov chain analysis. The probability of lysogeny is calculated by summing the probability of states that reach the highest level of *CI*. We compare our results with both experimental data and previous simulations performed by Arkin et al. on a complete master equation model. The experimental results are from Kourilsky (Kourilsky, 1973). Since it was not practical to measure the number of phages that infect any given cell, Kourilsky measured the fraction of cells that commit to lysogeny versus *average phage input* (API) (i.e., the proportion of phages to *E. coli* within the population). Kourilsky performed experiments for both "starved" *E. coli* and those in a "well-fed" environment. He found that the fraction that commits to lysogeny increases with increasing API, and that this fraction increases by more than an order of magnitude in a starved environment over a well-fed environment.

To map simulated MOI data onto API data, Arkin et al. used a Poisson distribution of the phage infections over the populations:

$$P(M, A) = \frac{A^M}{M!} e^{-A} \tag{20}$$

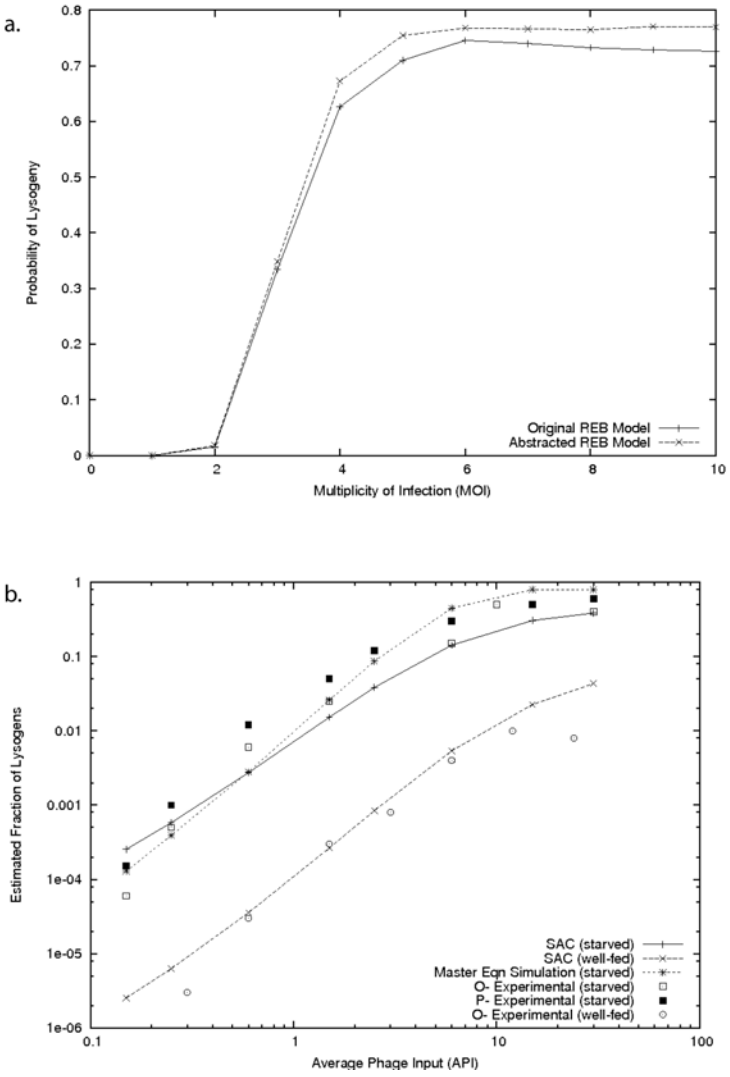
$$F_{lysogens}(A) = \sum_M P(M, A) \cdot F(M) \tag{21}$$

where M is the MOI, A is the API, and $F(M)$ is the probability of lysogeny determined by Markov analysis. We also used this method to map our MOI data. The results are shown in Figure 12(b). The individual points represent experimental measurements while the lines represent simulation results. Both the Arkin et al. simulation and our SAC model results track the starved data points reasonably well. Our SAC model results, however, are found in less than 7 minutes of computation time on a 3GHz Pentium4 with 1GB of memory. While modern computer technology and algorithmic improvements would greatly improve the simulation time of the Arkin et al. model, these results would still take several hours to generate on a similar computer to ours. Another notable benefit of our SAC method is that it can also produce simulation results for the well-fed case in about 7 minutes. These results could likely not be generated even today using the Arkin et al. master equation simulation method, since the number of simulation runs necessary is inversely proportional to the probability of lysogeny (i.e., about two orders of magnitude greater in the well-fed case than in the starved one).

CONCLUSION

This chapter presents a general methodology for systematically and automatically abstracting the complexities of large-scale biochemical reaction-based networks. The REB model abstractions significantly facilitate efficient temporal behavior analysis of such systems by substantially reducing the problem dimensionality in both species and reactions, thus potentially allowing for both simulation time ac-

Figure 12. Results from the phage λ decision circuit model. (a) Comparison of simulation results for the probability of lysogeny over MOI from the original model and its abstracted model where each data point has a margin of error of less than 0.01 with a 95 percent confidence. (b) Comparison of SAC results to experimental data.



celeration and computability gains while facilitating a high-level view of the network. To improve the numerical analysis time, a REB model can be further abstracted to a FSS model to allow for state space exploration based temporal behavior analysis approach on the underlying Markov chain. The system state space can be more aggressively reduced by transforming a REB model to a SAC model, enabling further improvement in the computational analysis time. Furthermore, since our approach allows for multiple levels of abstraction, it is broadly applicable to a wide range of biological systems and their representations—from CCK models to SCK models—including the gene regulatory networks upon which we have chosen to focus in this chapter.

The abstraction methods presented in this chapter coupled with a number of temporal behavior analysis methods are implemented in our modeling and analysis tool REB2SAC (Kuwahara et al., 2005, 2006a). By performing these transformations systematically and automatically and allowing an easily-configurable reduction control using REB2SAC, accuracy and efficiency of modeling biochemical systems at various levels of resolution can be significantly improved. Furthermore, to achieve better user experience of the tool, REB2SAC is integrated into a graphical-user-interface-based modeling and analysis tool called iBioSim, which can be downloaded from: (http://www.async.ece.utah.edu/iBioSim).

As a case study, we have illustrated an application of our model abstraction methodology to to systems-level analysis of a gene regulatory network. The preliminary results are promising. In this chapter, we have demonstrated that the probabilities of lysogeny for various MOI points obtained from the original phage λ circuit model can be approximated well by the results from the abstracted model with substantial computational gain. Furthermore, using the reduced FSS (SAC) model of the phage λ decision circuit, we are able to estimate the experimental results of the fraction of lysogens over API under various conditions (Kourilsky, 1973). The SAC model results are generated in a matter of minutes while the simulation results of REB models with a reasonable statistical confidence would have taken many hours to generate. Therefore, from this case study and others (Kuwahara et al., 2006b; Nguyen et al., 2007), we are able to: (1) ascertain the internal self-consistency of our approach by successfully cross-validating each abstraction level output against the results of the full underlying SCK model simulations; and (2) accurately estimate the biologically relevant properties, which typically require substantial numbers of hours of computation time via the original REB representation, yet could be computed in only minutes using our abstraction approach.

FUTURE RESEARCH DIRECTIONS

Advances in technologies such as DNA sequencing and gene expression profiling methods (Maxam and Gilbert, 1977; Schena et al., 1995) contribute to a increase in throughput for generation of data required for systematic approaches to understand gene regulatory networks. However, elucidating gene regulatory networks only via wet-lab experiments can be a daunting task, and complexity of this task increases proportional to the complexity of the network being analyzed. As more and more critical biological data become available and as the biological questions being addressed become more complex and challenging, the complexity of the systems of interests becomes so high that tackling such a problem only with wet-lab experiments eventually becomes infeasible. Thus, integration of computational methods with the process of biological research becomes more imminent.

Computational modeling and analysis can be applied to generate and screen hypotheses, which can stimulate the development of new experiments and effectively reduce the number of experiments to test hypotheses (Kitano, 2002; Collins et al., 2003). The first step of such a computational systems biology approach to the understanding of gene regulatory networks is to construct computational models encapsulating hypotheses and explaining experimental facts such as gene expression data. This can be done using various machine learning and data mining techniques (Barker et al., 2006; Friedman et al., 2000; Yu et al., 2004). Once quantitative computational models are constructed, they can be utilized to analyze the temporal behavior via simulation, allowing the hypothesis and assumptions encapsulated in each model to be analyzed and screened. Furthermore, a computational modeling and analysis approach comes with potentially unlimited controlling capabilities and abilities to capture virtually any

dynamical properties of the system, making possible a number of qualitative and quantitative analyses which cannot be done in wet-lab experiments. Since the quantitative data required to support systematic construction of such computational models are now becoming available via high-throughput molecular biology methods, this computational approach is now becoming possible and is able to provide useful biological insights (e.g., Borisuk and Tyson (1998); Edwards et al. (2001); Arkin et al. (1998); Wolf and Arkin (2002)). Furthermore, it can be used, for example, to apply an engineering approach to more efficiently and effectively analyze how a gene regulatory network can be controlled and designed to achieve specific functions (Brent, 2004; Arkin and Fletcher, 2006). In order to alleviate the complexity of such *in silico* analysis, further investigation on systematic and automatic model abstraction methodology coupled with a modeling language that accommodates multi-level representations of biochemical systems is crucial.

REFERENCES

Ackers, G. K., Johnson, A. D., & Shea, M. A. (1982). Quantitative model for gene regulation by λ phage repressor. *Proceedings of the National Academy of Sciences of the United States of America*, *79*, 1129–1133. doi:10.1073/pnas.79.4.1129

Arkin, A., & Fletcher, D. (2006). Fast, cheap, and somewhat in control. *Genome Biology*, *7*(8), 114. doi:10.1186/gb-2006-7-8-114

Arkin, A., Ross, J., & McAdams, H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells. *Genetics*, *149*, 1633–1648.

Ascher, U. M., & Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations*. SIAM.

Barker, N., Myers, C., & Kuwahara, H. (2006). Learning genetic regulatory network connectivity from time series data. In *The 19th International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems*.

Berry, R. S., Rice, S. A., & Ross, J. (2000). *Physical chemistry* (2nd edition). New York: Oxford University Press.

Borisuk, M. T., & Tyson, J. J. (1998). Bifurcation analysis of a model of mitotic control in frog eggs. *Journal of Theoretical Biology*, *195*(1), 69–85. doi:10.1006/jtbi.1998.0781

Brent, R. (2004). A partnership between biology and engineering. *Nature Biotechnology*, *22*, 1211–1214. doi:10.1038/nbt1004-1211

Briggs, G. E., & Haldane, J. B. S. (1925). A note on the kinetics of enzyme action. *The Biochemical Journal*, *19*, 339–339.

Cai, L., Friedman, N., & Xie, X. S. (2006). Stochastic protein expression in individual cells at the single molecule level. *Nature*, *440*(7082), 358–362. doi:10.1038/nature04599

Cao, Y., Gillespie, D., & Petzold, L. (2005a). Avoiding negative populations in explicit tau leaping. *The Journal of Chemical Physics*, 123.

Cao, Y., Gillespie, D., & Petzold, L. (2005b). The slow-scale stochastic simulation algorithm. *The Journal of Chemical Physics*, 122.

Cao, Y., Gillespie, D., & Petzold, L. (2006). Efficient stepsize selection for the tau-leaping method. *J. Chem. Phys.*

Cao, Y., Li, H., & Petzold, L. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting system. *The Journal of Chemical Physics*, *121*, 4059–4067. doi:10.1063/1.1778376

Cao, Y., & Petzold, L. (2005). Trapezoidal tau-leaping formula for the stochastic simulation of biochemical systems. In *Foundations of Systems Biology in Engineering*, 149–152.

Causton, H. C., Ren, B., Koh, S. S., Harbison, C. T., Kanin, E., & Jennings, E. G. (2001). Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the Cell*, *12*(2), 323–337.

Chatterjee, A., Vlachos, D. G., & Katsoulakis, M. A. (2005). Binomial distribution based tau-leap accelerated stochastic simulation. *The Journal of Chemical Physics*, *122*, 024112. doi:10.1063/1.1833357

Collins, F. S., Green, E. D., Guttmacher, A. E., & Guyer, M. S. (2003). A vision for the future of genomics research. *Nature*, *422*(6934), 835–847. doi:10.1038/nature01626

Davidson, E. H., Rast, J. P., Oliveri, P., Ransick, A., Calestani, C., & Yuh, C.-H. (2002). A genomic regulatory network for development. *Science*, *295*(5560), 1669–1678. doi:10.1126/science.1069883

DeRisi, J. L., Iyer, V. R., & Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, *278*(5338), 680–686. doi:10.1126/science.278.5338.680

Edwards, J. S., Ibarra, R. U., & Palsson, B. O. (2001). In silico predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. *Nature Biotechnology*, *19*, 125–130. doi:10.1038/84379

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*, 1183–1186. doi:10.1126/science.1070919

Finney, A., & Hucka, M. (2003). Systems biology markup language (SBML) level 2: Structures and facilities for model definitions. Retrieved from http://www.sbml.org/

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3–4), 601–620. doi:10.1089/106652700750050961

Gardiner, C. W. (2004). *Handbook of stochastic methods: For physics, chemistry, and the natural sciences*. Springer, 3rd edition.

Gibson, M., & Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, *104*, 1876–1889. doi:10.1021/jp993732q

Gillespie, D. (2000). The chemical langevin equation. *The Journal of Chemical Physics*, *113*(1). doi:10.1063/1.481811

Gillespie, D., & Petzold, L. (2003). Improved leap-size selection for accelerated stochastic simulation. *The Journal of Chemical Physics*, 119.

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics, 22*, 403–434. doi:10.1016/0021-9991(76)90041-3

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, *81*(25), 2340–2361. doi:10.1021/j100540a008

Gillespie, D. T. (1992a). *Markov processes an introduction for physical scientists*. Academic Press, Inc.

Gillespie, D. T. (1992b). A rigorous derivation of the chemical master equation. *Physica A, 188*, 404–425. doi:10.1016/0378-4371(92)90283-V

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, *115*(4), 1716–1733. doi:10.1063/1.1378322

Gillespie, D. T. (2005). *Handbook of materials modeling* (pp. 1735–1752). Springer.

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, *58*(1), 35–55. doi:10.1146/annurev.physchem.58.032806.104637

Golding, I., Paulsson, J., Zawilski, S. M., & Cox, E. C. (2005). Real-time kinetics of gene activity in individual bacteria. *Cell, 123*, 1025–1036. doi:10.1016/j.cell.2005.09.031

Guptasarma, P. (1995). Does replication-induced transcription regulate synthesis of the myriad low copy number proteins of *Escherichia coli? BioEssays*, *17*, 987–997. doi:10.1002/bies.950171112

Hermsen, R., Tans, S., & ten Wolde, P. R. (2006). Transcriptional regulation by competing transcription factor modules. *PLoS Computational Biology*, *2*(12), 164. doi:10.1371/journal.pcbi.0020164

Keener, J., & Sneyd, J. (1998). *Mathematical physiology*. Springer.

Kitano, H. (2002). Computational systems biology. *Nature, 420*(6912), 206–210. doi:10.1038/nature01254

Kourilsky, P. (1973). Lysogenization by bacteriophage lambda: I. multiple infection and the lysogenic response. *Molecular & General Genetics*, *122*, 183–195. doi:10.1007/BF00435190

Kuwahara, H. (2007). *Model abstraction and temporal behavior analysis of genetic regulatory networks*. Unpublished doctoral dissertation, University of Utah.

Kuwahara, H., & Myers, C. (2007). Production-passage-time approximation: A new approximation method to accelerate the simulation process of enzymatic reactions. In *The 11th Annual International Conference on Research in Computational Molecular Biology*.

Kuwahara, H., Myers, C., Barker, N., Samoilov, M., & Arkin, A. (2005). Asynchronous abstraction methodology for genetic regulatory networks. In *The Third International Workshop on Computational Methods in Systems Biology*.

Kuwahara, H., Myers, C., & Samoilov, M. (2006b). Abstracted stochastic analysis of type 1 pili expression in E. coli. In *The 2006 International Conference on Bioinformatics and Computational Biology*.

Kuwahara, H., Myers, C., Samoilov, M., Barker, N., & Arkin, A. (2006a). Automated abstraction methodology for genetic regulatory networks. *Trans. on Comput. Systematic Biology, VI*, 150–175.

Lodish, H., Berk, A., Zipursky, L. S., Matsudaira, P., Baltimore, D., & Darnell, J. (1999). *Molecular cell biology*. W. H. Freeman and Company.

Maxam, A. M., & Gilbert, W. (1977). A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America, 74*, 560–564. doi:10.1073/pnas.74.2.560

McAdams, H. H., & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(3), 814–819. doi:10.1073/pnas.94.3.814

McQuarrie, D. A. (1967). Stochastic approach to chemical kinetics. *Journal of Applied Probability*, *4*, 413–478. doi:10.2307/3212214

Munsky, B., & Khammash, M. (2006). The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, 124.

Myers, C. J., Belluomini, W., Killpack, K., Mercer, E., Peskin, E., & Zheng, H. (2001). Timed circuits: A new paradigm for high-speed design.

Newman, J. R. S., Ghaemmaghami, S., Ihmels, J., Breslow, D. K., Noble, M., DeRisi, J. L., & Weissman, J. S. (2006). Single-cell proteomic analysis of s. cerevisiae reveals the architecture of biological noise. *Nature*, *441*(7095), 840–846. doi:10.1038/nature04785

Nguyen, N., Kuwahara, H., Myers, C., & Keener, J. (2007). The design of a genetic muller C-element. In *The 13th IEEE International Symposium on Asynchronous Circuits and Systems*.

Pedraza, J. M., & van Oudenaarden, A. (2005). Noise propagation in gene networks. *Science, 307*(5717), 1965–1969. doi:10.1126/science.1109090

Peleš, S., Munsky, B., & Khammash, M. (2006). Reduction and solution of the chemical master equation using time scale separation and finite state projection. *The Journal of Chemical Physics*, 125.

Ptashne, M. (1992). *A genetic switch*. Cell Press & Blackwell Scientific Publishing.

Rao, C. V., & Arkin, A. P. (2003). Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. *Journal of Physical Chemistry*, *118*(11).

Rao, C. V., Wolf, D. M., & Arkin, A. P. (2002). Control, exploitation, and tolerance of intracellular noise. *Nature*, *420*, 231–238. doi:10.1038/nature01258

Raser, J. M., & O'Shea, E. K. (2005). Noise in gene expression: Origins, consequences, and control. *Science*, *309*(5743), 2010–2013. doi:10.1126/science.1105891

Rathinam, M., Cao, Y., Petzold, L., & Gillespie, D. (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, *119*, 12784–12794. doi:10.1063/1.1627296

Samoilov, M. (2003). Stochastic effects in enzymatic biomolecular systems: Framework, fast and slow species, and quasi-steady state approximations. In *Workshop on Dynamical Stochastic Modeling in Biology*.

Samoilov, M., Plyasunov, S., & Arkin, A. P. (2005). Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences US*, *102*(7), 2310–2315. doi:10.1073/pnas.0406841102

Samoilov, M. S., & Arkin, A. P. (2006). Deviant effects in molecular reaction pathways. *Nature Biotechnology*, *24*, 1235–1240. doi:10.1038/nbt1253

Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, *270*(5235), 467–470. doi:10.1126/science.270.5235.467

Segel, Lee A., & Slemrod, M. (1989). The quasi-steady-state assumption: A case study in perturbation. *SIAM Review*, *31*(3), 446–477. doi:10.1137/1031091

Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains*. Princeton University Press.

Tian, T., & Burrage, K. (2004). Binomial leap methods for simulating stochastic chemical kinetics. *The Journal of Chemical Physics*, *121*, 10356–10364. doi:10.1063/1.1810475

van Kampen, N. G. (1992). *Stochastic processes in physics and chemistry*. Elsevier.

Wolf, D. M., & Arkin, A. P. (2002). Fifteen minutes of *fim*: Control of type 1 pili expression in *E. coli*. *OMICS: A Journal of Integrative Biology*, *6*(1), 91–114. doi:10.1089/15362310252780852

Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J., & Jarvis, E. D. (2004). Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics (Oxford, England)*, *20*, 3594–3603. doi:10.1093/bioinformatics/bth448

ADDITIONAL READING

Ackers, G. K., Johnson, A. D., & Shea, M. A. (1982). Quantitative model for gene regulation by λ phage repressor. *Proceedings of the National Academy of Sciences of the United States of America*, *79*, 1129–1133. doi:10.1073/pnas.79.4.1129

Arkin, A., & Fletcher, D. (2006). Fast, cheap, and somewhat in control. *Genome Biology*, *7*(8), 114. doi:10.1186/gb-2006-7-8-114

Arkin, A., Ross, J., & McAdams, H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. *Genetics*, *149*, 1633–1648.

Brent, R. (2004). A partnership between biology and engineering. *Nature Biotechnology*, *22*, 1211–1214. doi:10.1038/nbt1004-1211

Cao, Y., Gillespie, D., & Petzold, L. (2005). The slow-scale stochastic simulation algorithm. *The Journal of Chemical Physics,* 122.

Cao, Y., Li, H., & Petzold, L. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting system. *The Journal of Chemical Physics*, *121*, 4059–4067. doi:10.1063/1.1778376

Cao, Y., & Petzold, L. (2005). Trapezoidal tau-leaping formula for the stochastic simulation of biochemical systems. In *Foundations of Systems Biology in Engineering*, 149–152.

Collins, F. S., Green, E. D., Guttmacher, A. E., & Guyer, M. S. (2003). A vision for the future of genomics research. *Nature*, *422*(6934), 835–847. doi:10.1038/nature01626

Davidson, E. H., Rast, J. P., Oliveri, P., Ransick, A., Calestani, C., & Yuh, C.-H. (2002). A genomic regulatory network for development. *Science*, *295*(5560), 1669–1678. doi:10.1126/science.1069883

Edwards, J. S., Ibarra, R. U., & Palsson, B. O. (2001). In silico predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. *Nature Biotechnology*, *19*, 125–130. doi:10.1038/84379

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*, 1183–1186. doi:10.1126/science.1070919

Fall, C., Marland, E., Wagner, J., & Tyson, J. (Eds.). (2002). *Computational cell biology*. Springer.

Frazier, M. E., Johnson, G. M., Thomassen, D. G., Oliver, C. E., & Patrinos, A. (2003). Realizing the potential of the genome revolution: The genomes to life program. *Science*, *300*(5617), 290–293. doi:10.1126/science.1084566

Gardiner, C. W. (2004). *Handbook of stochastic methods: For physics, chemistry, and the natural sciences*. Springer, 3rd edition.

Gibson, M., & Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, *104*, 1876–1889. doi:10.1021/jp993732q

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, *22*, 403–434. doi:10.1016/0021-9991(76)90041-3

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, *81*(25), 2340–2361. doi:10.1021/j100540a008

Gillespie, D. T. (1992). *Markov processes an introduction for physical scientists*. Academic Press, Inc.

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics*, *115*(4), 1716–1733. doi:10.1063/1.1378322

Gillespie, D. T. (2005). *Handbook of materials modeling* (pp. 1735–1752). Springer.

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, *58*(1), 35–55. doi:10.1146/annurev.physchem.58.032806.104637

Jong, H. D. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*(1), 67–103. doi:10.1089/10665270252833208

Keener, J., & Sneyd, J. (1998). *Mathematical physiology*. Springer.

Kierzek, A. M., Zaim, J., & Zielenkiewicz, P. (2001). The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *The Journal of Biological Chemistry*, *276*, 8165. doi:10.1074/jbc.M006264200

Kincaid, D., & Cheney, W. (1996). *Numerical analysis*. Brooks/Cole Publishing Company, 2nd edition.

Kitano, H. (2002a). Computational systems biology. *Nature*, *420*(6912), 206–210. doi:10.1038/nature01254

Kitano, H. (2002b). Systems biology: A brief overview. *Science*, *295*, 1662–1664. doi:10.1126/science.1069492

Kuwahara, H. (2007). *Model abstraction and temporal behavior analysis of genetic regulatory networks*. Unpublished doctoral dissertation, University of Utah.

Kuwahara, H., & Myers, C. (2007). Production-passage-time approximation: A new approximation method to accelerate the simulation process of enzymatic reactions. In *The 11th Annual International Conference on Research in Computational Molecular Biology*.

Kuwahara, H., Myers, C., Barker, N., Samoilov, M., & Arkin, A. (2006). Automated abstraction methodology for genetic regulatory networks. *Trans. on Comput. Systematic Biology*, *VI*, 150–175.

Lodish, H., Berk, A., Zipursky, L. S., Matsudaira, P., Baltimore, D., & Darnell, J. (1999). *Molecular cell biology*. W. H. Freeman and Company.

Maheshri, N., & O'Shea, E. K. (2007). Living with noisy genes: How cells function reliably with inherent variability in gene expression. *Annual Review of Biophysics and Biomolecular Structure*, *36*(1), 413–434. doi:10.1146/annurev.biophys.36.040306.132705

McAdams, H. H., & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(3), 814–819. doi:10.1073/pnas.94.3.814

McAdams, H. H., & Arkin, A. (1998). Simulation of prokaryotic genetic circuits. *Annual Review of Biophysics and Biomolecular Structure*, *27*, 199–224. doi:10.1146/annurev.biophys.27.1.199

McAdams, H. H., & Shapiro, L. (1995). Circuit simulation of genetic networks. *Science*, *269*(5224), 650–656. doi:10.1126/science.7624793

McQuarrie, D. A. (1967). Stochastic approach to chemical kinetics. *Journal of Applied Probability*, *4*, 413–478. doi:10.2307/3212214

Munsky, B., & Khammash, M. (2006). The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, 124.

Pedraza, J. M., & van Oudenaarden, A. (2005). Noise propagation in gene networks. *Science*, *307*(5717), 1965–1969. doi:10.1126/science.1109090

Peleš, S., Munsky, B., & Khammash, M. (2006). Reduction and solution of the chemical master equation using time scale separation and finite state projection. *The Journal of Chemical Physics*, 125.

Press, W. H., Flannery, B. P., Teukolsky, S. A., & Vetterling, W. T. (1992). *Numerical recipes in c: The art of scientific computing*. Cambridge University Press, 2nd edition.

Ptashne, M. (1992). *A genetic switch*. Cell Press & Blackwell Scientific Publishing.

Rao, C. V., & Arkin, A. P. (2003). Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. *Journal of Physical Chemistry*, *118*(11).

Rao, C. V., Wolf, D. M., & Arkin, A. P. (2002). Control, exploitation, and tolerance of intracellular noise. *Nature*, *420*, 231–238. doi:10.1038/nature01258

Raser, J. M., & O'Shea, E. K. (2004). Control of stochasticity in eukaryotic gene expression. *Science*, *304*, 1811–1814. doi:10.1126/science.1098641

Rathinam, M., Cao, Y., Petzold, L., & Gillespie, D. (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, *119*, 12784–12794. doi:10.1063/1.1627296

Samoilov, M. (2003). Stochastic effects in enzymatic biomolecular systems: Framework, fast and slow species, and quasi-steady state approximations. In *Workshop on Dynamical Stochastic Modeling in Biology*.

Samoilov, M. S., & Arkin, A. P. (2006). Deviant effects in molecular reaction pathways. *Nature Biotechnology*, *24*, 1235–1240. doi:10.1038/nbt1253

Shea, M. A., & Ackers, G. K. (1985). The OR control system of bacteriophage lambda: A physical-chemical model for gene regulation. *Journal of Molecular Biology*, *181*, 211–230. doi:10.1016/0022-2836(85)90086-5

Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains*. Princeton University Press.

van Kampen, N. G. (1992). *Stochastic processes in physics and chemistry*. Elsevier.

KEY TERMS AND DEFINITIONS

Finite State System Model Transformation: Model abstraction to transform an REB model to a FSS model by introducing upper limit molecule count for each species.

FSS Model: A model whose state space is finite.

Gene Regulatory Network: A network that regulates expression of certain genes.

Model Abstraction: Transformation of a model to ease analysis of model properties of interest.

N-ary Transformation: Model abstraction to transform an REB model to a FSS model by capturing the state of each species with nary-level.

Reaction-Based Abstraction: Model abstraction to transform an REB model to another REB model.

REB Model: A model based on the reaction connectivity of species.

SCK: Stochastic-discrete formalism to model temporal behavior of chemically reacting systems.

Chapter 16
Improved Model Checking Techniques for State Space Analysis of Gene Regulatory Networks

Hélio C. Pais
Cadence Research Laboratories, USA and INESC-ID/IST, Portugal

Kenneth L. McMillan
Cadence Research Laboratories,USA

Ellen M. Sentovich
Cadence Research Laboratories,USA

Ana T. Freitas
INESC-ID/IST, Portugal

Arlindo L. Oliveira
Cadence Research Laboratories, USA and INESC-ID/IST, Portugal

ABSTRACT

A better understanding of the behavior of a cell, as a system, depends on our ability to model and understand the complex regulatory mechanisms that control gene expression. High level, qualitative models of gene regulatory networks can be used to analyze and characterize the behavior of complex systems, and to provide important insights on the behavior of these systems. In this chapter, we describe a number of additional functionalities that, when supported by a symbolic model checker, make it possible to answer important questions about the nature of the state spaces of gene regulatory networks, such as the nature and size of attractors, and the characteristics of the basins of attraction. We illustrate the type of analysis that can be performed by applying an improved model checker to two well studied gene regulatory models, the network that controls the cell cycle in the yeast S. cerevisiae, and the network that regulates formation of the dorsal-ventral boundary in D. melanogaster. The results show that the insights provided by the analysis can be used to understand and improve the models, and to formulate hypotheses that are biologically relevant and that can be confirmed experimentally.

DOI: 10.4018/978-1-60566-685-3.ch016

INTRODUCTION

The next advances in the field of systems biology are critically dependent on a better understanding of gene regulatory networks, and abstract models are essential in order to improve our understanding of the complex mechanisms underlying gene regulation.

Indeed, the highly complex transcriptional regulation of gene expression in eukaryotes occurs through the coordinated action of multiple transcription factors, and can be understood in detail only if one views the system as a network.

Regrettably, the complex physical and biochemical mechanisms involved in gene regulation make it difficult to identify regulatory mechanisms directly from first principles, and even sophisticated experimental methods are difficult to apply directly to the identification of the parameters of gene regulatory networks. For these reasons, computational approaches that help researchers identify and analyze genetic regulatory networks are essential, and represent a fundamental tool in our quest for the understanding of organisms as biological systems.

Gene regulatory networks, as well as other biological networks, have been modeled and studied using a variety of levels of abstraction. At the more detailed levels, ordinary differential equations (ODE) that relate different concentration levels within the cells have been used to model the dynamics of some small, well understood systems. Less detailed qualitative models have been extensively used to perform analysis of more complex systems [de Jong et al., 2003, Chabrier and Fages, 2003, Li et al., 2004, Garg et al., 2007], and have the advantage that they require less knowledge of the exact parameters governing the dynamics of the system. In qualitative models each of the components of the system (gene, protein, metabolite, etc.) is represented as a variable with a finite domain, and each variable is updated in accordance with a discrete transition function, that specifies its value as a function of the present value of the variables of the system.

Qualitative models have a number of significant advantages over quantitative ones. They can be used to analyze larger systems and, perhaps more importantly, existing tools, such as model checkers, can be used to analyze not only a particular trajectory of the system in state space, but also to characterize the state space as a whole. Understanding the characteristics of the state space can lead to important insights on how these networks have evolved and are wired. One of the problems of using qualitative models is the large number of behaviors that are possible, when one is dealing with complex systems whose dynamics cannot be sufficiently constrained. Model checking techniques have been proposed to deal with this problem.

Verification of biological network properties based on model checking provides a powerful method to analyze models of molecular interaction networks [Shults and Kuipers, 1997]. Using this methodology coerces the user to formulate interesting questions, and to interpret the answers, something that is especially difficult when dealing with very large models. The problem of posing relevant questions is critical in model assessment, in general, but even more so when using model checking. For instance, a property like **once the concentration of protein P_1 reaches some threshold, the concentration of protein P_2 will start to increase only after some reaction R has stopped**, corresponds to the CTL formula $AG(P_1 \rightarrow ((EFP_2) \wedge \neg E\,[RUP_2\,]))$.

The next sections describe how a model checker that includes some extra functionality can be used to analyze the characteristics of the state space of a gene regulatory network, and obtain additional insight about the behavior of the biological system.

RELATED WORK

Gene Regulatory Networks

One very general way to construct abstractions of biological systems is to model them as networks, or graphs [Hasty et al., 2001]. In the case of gene regulatory networks, a gene (and the corresponding protein) can be viewed as a node in this network. Edges between nodes correspond to interactions. An example of a common interaction is transcription regulation, where the abundance of a given transcription factor (a protein) affects directly the level of expression of the target gene. Associated with each node is a function of its inputs. In quantitative models, the value of the variable that corresponds to a given node obeys a specific equation, typically an ordinary differential equation, that specifies its derivative as a function of the values of the other variables that are inputs to this function. In the case of qualitative models, that are the subject of this work, the function at a node is a discrete function of its inputs.

In both cases, these functions can be viewed as performing information processing in the cell, determining cellular behavior. Graph models of gene regulation networks have been developed and extensively used to make predictions of the behavior of the cell.

An interesting parallel can be established with logic circuits, that are at the heart of modern computers and other digital devices. Logic circuits are also modeled and analyzed as networks [de Micheli, 1994]. Each node in a logic network represents either a logic gate or a register, and this network defines the dynamics, and, ultimately, the functionality of the system.

It is therefore not surprising that tools that have been developed mostly to analyze the behavior of digital electrical networks can be applied to the analysis of gene regulatory networks. Among these tools are simulators and model checkers.

Model Checking

Model checking is a method originally developed for the analysis of concurrent programs, although it can in principle be applied to any discrete dynamical system. The method makes it possible to determine whether the system exhibits a given temporal property, that is, a property that specifies some desired structure in the transition sequences of the system. Properties are specified using a specialized logical notation called temporal logic. This notation makes it possible to state formally assertions such as “if x occurs then inevitably y will occur in the future”, or “it is always possible that z will occur in the future”.

Kripke Structures

Mathematically, a discrete dynamical system is modeled as a *directed graph*. This is a pair (*S*,*R*), consisting of a set of *states S* and a set of *transitions R*. We will consider only the case where the set of states *S* is finite. Pictorially, we represent the states as circles and the transitions as arrows connecting the circles, as shown in Figure 5. More precisely, a transition is a pair (*s*,*t*) where *s* and *t* are states. A transition from state *s* to state *t* indicates that the system may evolve directly from state *s* to state *t*. Note that this is a qualitative model of the system’s behavior. There is no quantitative probability or rate associated with a transition. There may be transitions from state *s* to multiple states. In this case, the behavior of the model is non-deterministic.

A *path* in a graph (*S*,*R*) is a sequence s_0,s_1, of states from *S*, such that each consecutive pair of states (s_i,s_{i+1}) is a transition in *R*. Informally, a path is obtained by starting at some state and following the arrows. A path can be either finite or infinite, and represents a possible temporal evolution of the system.

In order to make meaningful statements about paths, we must be able to make distinctions between states. To do this, we introduce *labels* on the states. A label is just a symbol whose presence represents some fact about the system state. For example, the letter *P* might represent the presence of some chemical species, or the binding of a protein to a given receptor site. Note again that this is qualitative information, and cannot tell us, for example, the concentration of a given species, except perhaps to say that it is above or below a given threshold. In logical terms, these labels are referred to as *atomic propositions*, since they are logical propositions that cannot be decomposed into more primitive propositions.

A system model thus becomes a labeled directed graph (*S*,*R*,*L*), where *S* and *R* form a graph, and *L* is a *labeling function* that associates some set of labels or atomic propositions with every state of *S*. This structure is referred to as a Kripke structure, after philosopher and logician Saul Kripke, who first gave a mathematical interpretation to the class of logics we are concerned with using this kind of structure. More precisely, the Kripke structure is obtained by unwinding the labeled directed graph into an infinite tree.

Specifying Properties in CTL

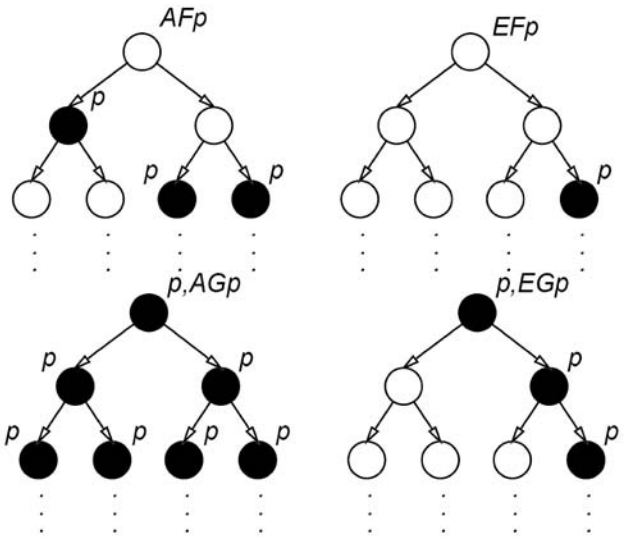
The logic CTL (Computation Tree Logic) provides a way to make statements about the structure of paths in the Kripke model [Clarke and Emerson, 1981]. It provides a simple set of operators for making statements about paths and states. Suppose, for example, that *p* is a fact that is true of some subset of the system states. The formula *Fp* is true of those paths that contain *some* state in which *p* is true. We read this as "eventually *p*", since it indicates that if one follows the path far enough, a state satisfying *p* will be reached. The formula *Gp* is true of those paths in which *p* is true in *every* state. We read this as "always *p*". Two additional operators called *path quantifiers* allow us to make statements about the set of all paths emanating from a given state. Suppose that *q* is a fact about paths. Then *Aq* is true in a state when *all* paths starting from that state satisfy *q*. The formula *Eq* is true in a state when *some* path starting from that state satisfies *q*.

In CTL every path operator, such as *F* or *G* must be preceded by a path quantifier *A* or *E*, indicating whether the formula applies to all paths or some path from a given state. Thus, all CTL formulas are facts about states, not paths. As an example *AFp* means "along all paths, eventually *p*", or "inevitably *p*." We can put the operators together to make more complex statements. As an example, the formula *AG EFp* means "always, along all paths, a state satisfying *p* may be reached", or alternatively "*p* is never ruled out." We can also combine facts about states with the standard connectives of propositional (or Boolean) logic. For example, the formula $p \rightarrow q$ is read as "if *p* then *q*", or "*p* implies *q*". Thus $AG(p \rightarrow AFq)$ means "whenever *p* happens, *q* inevitably occurs". We can also apply $\vee$ (logical *or*), $\wedge$ (logical *and*) and $\neg$ (logical *not*). Figure 1 shows some examples of states (shown in black) satisfying various CTL formulas.

In addition to the above described forms, we have *Xp*, which is true of a path when *p* holds in the second state (at the "next time") and *pUq* which is true of a path when *q* holds in some state, and *p* holds in all the preceding states ("*p* until *q*").

The syntax of CTL is given by the following grammar, where *S* represents the state formulas and *A* the atomic propositions:

Figure 1. Examples of states satisfying various CTL formulas



$S ::= \text{false} | A | \neg S | S \vee S | EXS | AXS | E(S \; U \; S) | A(S \; U \; S)$ (1)

The forms not shown above can be expressed as abbreviations. For example, $p \wedge q$ abbreviates $\neg(p \vee \neg q)$ and *EFp* abbreviates *E*(true *U p*).

Checking Formulas

Given a fact about states expressed as a CTL formula φ, a natural question to ask is "is ϕ true in state s_0 ? " or more generally "in what states is ϕ true? " This is called the *model checking problem*. The model checking problem can be solved by a very simple graph algorithm that moves from inner to outer subformulas. Suppose, for example, that we are given the formula *EF*ϕ, where ϕ is an arbitrary formula. We first compute the set of states in which φ is true, which we will also denote ϕ. We then compute the following sequence of sets:

$$R_0 = \phi \quad (2)$$

$$R_1 = R_0 \cup pre(R_0) \quad (3)$$

$$R_2 = R_1 \cup pre(R_1) \quad (4)$$

$$\ldots \quad (5)$$

where, for a given set of states *Q*, pre(*Q*) is the set of states having a transition to any state in *Q*. Intuitively, R_i is the set of states that can reach a state satisfying ϕ in *i* steps or fewer. The set of states satisfying *EF*ϕ, that is to say those that can reach ϕ in any number of steps, is given by the union of the R_i. Since the number of states is finite, this union is the stable limit of the sequence, which we obtain when

$R_{i+1}=R_i$. A similar characterization of the states satisfying $EG\phi$ is obtained by replacing $\cup$ with $\cap$ in the above recurrence, and the other CTL operators have similarly simple characterizations (for example, the set satisfying $p \vee q$ is just $p \cup q$ and $\neg p$ is the set complement $S|p$). This allows us to make a sort of calculator that computes the set of states satisfying a given CTL formula much in the way one would evaluate a simple algebraic formula.

This approach, described in a seminal paper by Clarke and Emerson [Clarke and Emerson, 1981] gives an algorithm for determining the set of states of a finite Kripke model satisfying any given CTL formula. The running time is proportional to the size of the formula and the square of the size of the model. Although a linear time algorithm is possible, this simple approach is often used in practice because it lends itself to the use of highly efficient representations of sets.

SMV

A large number of model checkers has been proposed and made available. When the model checker uses an implicit representation of sets (see following section), the process is known as *symbolic model checking* [McMillan, 1993]. SMV is a well known symbolic model checking program. In SMV, the network is described in the SMV language [McMillan, 1999], which can be divided roughly into three parts: declarations, structure, and expressions.

- The definitional part of the language declares signals and their relationship to each other. It includes type declarations and assignments.
- The structural part of the language combines definitional components. It provides language constructs for defining modules and structured data types to instantiate them. It also provides constructor loops, for describing regularly structured systems, and a collection of conditional structures that make describing complicated state transition tables easier.
- The language of expressions in SMV is very similar to expressions in other languages, and is used to combine primitive variables corresponding to signals.

Examples of the SMV language will be given in the later sections. This tool was originally developed at Carnegie Mellon University, but newer variants of the system, using the same language, are also available, such as nuSMV or CadenceSMV.

Binary Decision Diagrams

The need for efficient representations of sets becomes clear when we consider that the number of states of even relatively small systems can be astronomically large. For example, suppose the state of a system is described by N binary variables (for example, the presence or absence of bound proteins at N receptor sites). Then the number of states of the system is potentially 2^N. If N is even moderately large (say 30 or 40) then the set of states becomes too large to represent in computer memory. This has been referred to as the *state explosion problem*.

One approach to this problem is to use an *implicit* representation of sets that is capable of characterizing a very large set with a relatively small structure by exploiting some regularity or redundancy in the set. If the operations pre, $\cup$ and set complement can be efficiently computed on the representation, then it is suitable for the application.

Although many representations are used in symbolic model checking, the most common one for finite-state systems is the Reduced Ordered Binary Decision Diagram (ROBBD), which is often referred to as simply a BDD [Bryant, 1986]. We assume that the state of the system is characterized by N binary variables $x_1 \ldots x_N$. A set of states can thus be characterized by a decision tree. This tree consists of decision nodes, each of which is labeled with a variable x_i and has two outgoing branches. The left branch is taken in case the variable x_i is false, and the right branch in case it is true. In addition, the tree contains leaf nodes labeled 0 or 1. To evaluate the decision tree, one begins at the unique root node, and follows the branches according to the truth values of the variables labeling them. If, for a given state, one arrives at a leaf labeled 1, then that state is in the set, otherwise not. Figure 2 shows an example of a decision tree for the set of states satisfying the simple Boolean formula $ab \vee cd$, with variables a,b,c,d. In this tree, subtrees at nodes marked * are identical, while node marked + is redundant.

A decision tree is said to be ordered, if along any path from the root to a leaf, each variable occurs once, in a fixed order. Notice that the tree of Figure 2 is ordered a,b,c,d. Notice also that this tree contains a certain amount of redundancy. For example, the tree contains several subtrees that are identical. Moreover, there are decision nodes that are redundant, in the sense that the two branches arrive at identical subtrees. We can eliminate this redundancy by combining identical subtrees into a single subtree and removing the redundant decision nodes. The result of this transformation is shown in Figure 3. Notice that the structure is smaller, and that it is no longer a tree because of the convergent paths (technically it is a directed acyclic graph, or DAG)). This structure is referred to as a Reduced Ordered Binary Decision Diagram (ROBDD).

Because an ROBDD eliminates a certain kind of redundancy in the representation of a set, it can sometimes compactly represent a very large set of states. Moreover, there are efficient ROBDD algorithms to compute the set operations needed for model checking: union, complement and pre [Bryant, 1986, McMillan, 1993]. The last is based on representing the set of model transitions R implicitly as a BDD. Thus, we can build our CTL calculator entirely with operations on ROBDD's, avoiding any explicit enumeration of sets of states. This method sometimes allows us to handle astronomically large state sets, typically on the order of 10^{15} to 10^{20} states.

Figure 2. Ordered decision tree

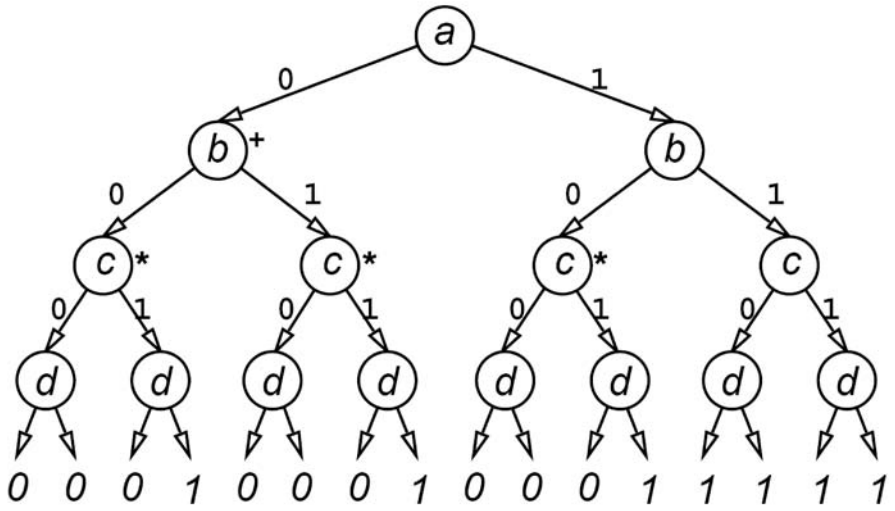
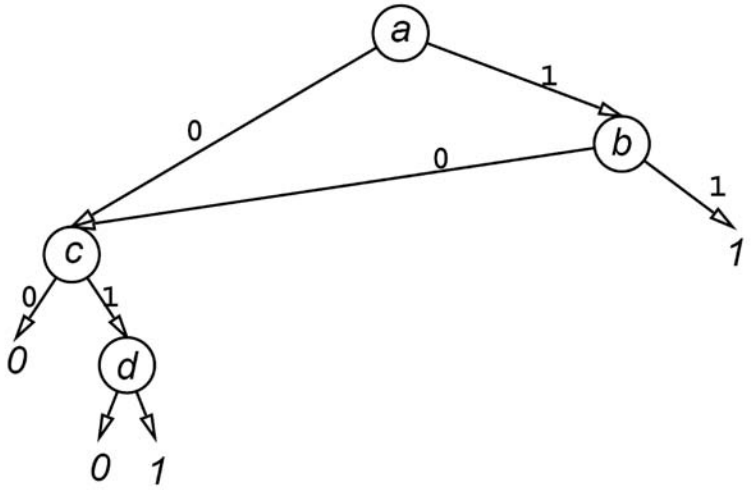


Figure 3. Reduced ordered decision diagram



There are other approaches to handling large state spaces in model checking, for example, methods based on Boolean satisfiability solvers [Biere et al., 1999, McMillan, 2003]. However, the advantage of the BDD-based symbolic model checking method for our purposes is that it allows us to compute compact representations of biologically interesting sets of states. This allows us to extract information from these sets, such as the number of states, and the dependence of the set on certain variables. We will also observe that BDD-based methods can be used to construct sets of states not expressible in CTL such as those representing "steady states" of the system (technically, the terminal strongly connected components of the graph).

Tools for Qualitative Analysis of Biological Networks

A number of tools have been developed specifically for the analysis of qualitative models. Among these tools, Biocham [Fages et al., 2004], GenYsis [Garg et al., 2007], GinSim [Gonzalez et al., 2006b], and GNA [de Jong et al., 2003] are relevant for the type of analysis described in this chapter.

Biocham is targeted mainly at biochemical models, and is limited in the types of descriptions that it accepts. GenYsis can accept, in principle, any discrete models and performs implicit computation of steady states using BDDs to represent the transition function of the system. However, it does not use the most efficient algorithm for steady state computation, and does not make available any interface with a general model checker.

Both GinSim and GNA accept arbitrary discrete models, but perform explicit traversal of the state transition graph, and cannot, therefore, be used to analyze the state spaces of larger systems. BioCham and GNA have the ability to interface with a model checker, by exporting SMV files.

Our improved model checking tool is based on NuSMV, a tool that, as referred, uses implicit traversal of the state space using BDDs. This makes it able to manipulate networks that are many orders of magnitude larger than those that can be handled using explicit enumeration methods.

CHECKING STATE SPACE PROPERTIES OF GENE REGULATORY NETWORKS

Standard model checking tools can be used to analyze qualitative models of biological phenomena. Indeed, GNA and Biocham can be used in conjunction with a model checker. It is then possible to write properties of the system in a temporal logic language and use the model checker to verify if the model satisfies the properties.

However, the functionality commonly available in standard model checkers is not sufficient to perform some of the analyzes that are, in many cases, required, when this approach is being used to study state space properties of biological networks.

One reason is that temporal logics most commonly available in model checkers, like CTL, lack the expressive power to formulate questions related to the steady state behavior of the system. The knowledge of the steady states of the model is fundamental to understand if the model is reproducing the behavior of the biological system. Tools like GNA and GinSim compute the steady states, but do it by explicitly traversing the state transition graph. Therefore, they can only be applied to relatively small systems.

Another reason is that standard model checkers can only tell if a property is or not verified over all reachable states. This is usually enough if one is verifying human-designed hardware systems. However, in the study of biological systems, it may be useful to have more information, namely what fraction of states verify the given property. This number can serve as a rough measure of the robustness of the model.

Network Model

The tool we developed, BioNuSMV, which integrates the ideas described in this chapter, accepts a specification described in a high level language, and is, therefore, very flexible. The next sections describe the specification language and the type of analyzes supported by this tool. BioNuSMV is based on the NuSMV software [Cimatti et al., 2002], an open source tool for symbolic model checking.

BioNuSMV uses a simple formalism to represent gene regulatory networks. A set of n variables x_i, that can take values in D_i, keep the state of the system, and correspond to the nodes in the network graph.

The state of a node i at time t is represented by the variable $x_i(t)$. The value of x_i at time $t+1$ is given by an arbitrary function of the values of all variables in time t.

$$x_i(t+1) = f_i(x_1,..,x_i,...,x_n) \quad (6)$$

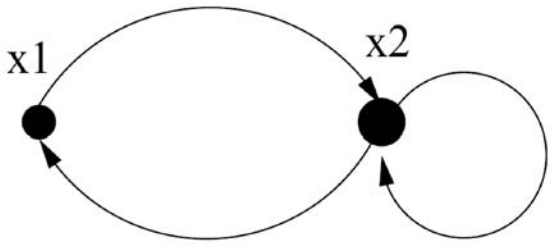
The transition between states of the system can be either synchronous or asynchronous, depending on whether the updates are performed all in the same module, or in separate modules. Intermediate cases (e.g., partially synchronous models) can also be specified.

For example, consider the network presented in Figure 4, with $D_1 = D_2 = \{false,true\}$. In this network, gene x_1 regulates gene x_2, while gene x_2 auto-regulates itself and also regulates gene x_1.

The state of gene x_1 follows the state of gene x_2 (with a time delay) and gene x_2 becomes active if either x_1 or x_2 are active in the previous instant of time. The values of the two variables are therefore updated according to following equations:

$$x_1(t+1) = x_2(t) \quad (7)$$

Figure 4. Example network with two genes



$$x_2(t+1) = x_1(t) \vee x_2(t) \quad (8)$$

The synchronous and asynchronous update schemes will generate the transition graphs presented, respectively, in figures 5.a and 5.b. In this figure, states are encoded with two bits, that correspond to the values of the activation of genes x_1 and x_2.

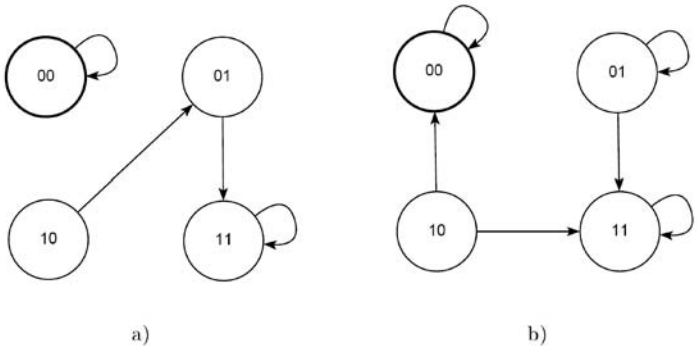
The synchronous version of this system could be translated to the SMV language as follows:

```
MODULE main
VAR
 x1 : boolean;
 x2 : boolean:
ASSIGN
 init(x1) := {TRUE,FALSE};
 init(x2) := {TRUE,FALSE};
 next(x1) := x2;
 next(x2) := x1 | x2;
```

The next SMV example shows how the boolean state variable sbf_v is changed in accordance with the values of the variables Cln3 and Clb1,2. These variables correspond to the SBF, Cln3 and Clb1,2 genes, involved in the yeast cell cycle regulatory network. This network model, depicted in figure 6 will be analyzed and studied in section 4.

```
MODULE sbf_m(cln3,clb12)
VAR
  sbf_v : boolean;
ASSIGN
  init(sbf_v) := 0;
  next(sbf_v) := case
    cln3 - clb12 >0 : 1;
    cln3 - clb12 <0 : 0;
    cln3 - clb12 =0 : sbf_v;
  esac;
```

Figure 5. State transition diagram for the synchronous (a) and asynchronous (b) update cases. Each state is encoded with two bits, that represent the values taken by state variables x_1 and x_2.



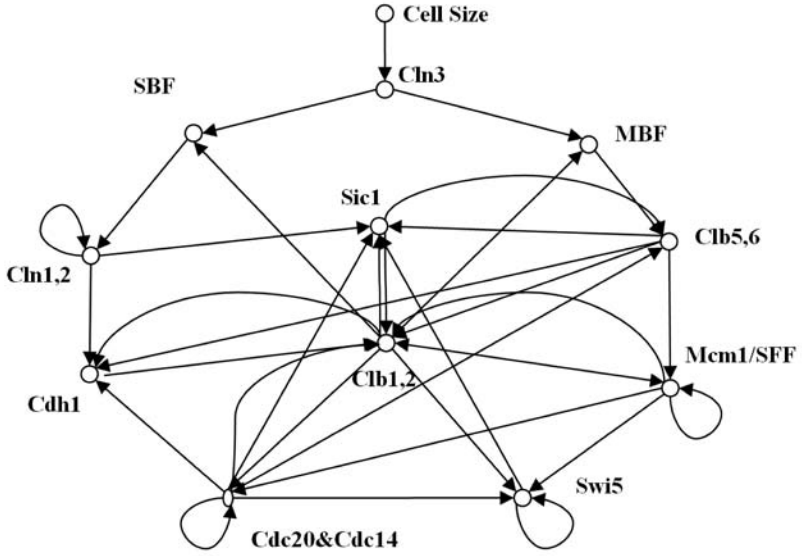

```
--next(sbf_v):=0;
```

With SMV it is possible to construct a model synchronous, asynchronous or hybrid. The use of SMV modules also makes the codification of multi-cellular models easier, as the transition rules for each component of the model have to be written only once. To use the same component in different cells it is only necessary to declare an instance of it for each cell.

For the two gene example system of Figure 5 a), we may, for example, want to address the following biological questions:

- If at least one gene is active, will both genes be active in the future? This means that all states in which at least one gene is active will always reach the state where both genes are active, and,

Figure 6. Discrete model for the yeast cell cycle (from [Li et al., 2004])



therefore, that the formula, should be verified by the system. Since this is true for this example, the answer to the biological question is positive.

- Will there always be a sequence of events that leads to a state where both genes are inactive? This means that there should be always one path that will reach the state where both genes are inactive. The formula is not true in general, and is true only for state *(false,false)*. Therefore, the answer to the biological question is negative.

State Space Analysis in BioNuSMV

Most model checkers, including NuSMV, work by verifying if the specified properties hold for the initial state of the system, or for a set of specified states. However, as mentioned previously, standard model checkers lack some features that are useful when analyzing discrete models of biological systems.

BioNuSMV implements these features, which make it possible to perform analyzes that enable researchers to gain additional insights into the characteristics of the state space of gene regulatory networks.

The new functionalities of BioNuSMV, relative to NuSMV and other standard model checkers are:

1. When a property is not satisfied by the network under analysis, the model checker outputs not only a counter-example, but also the fraction of states that do not satisfy the property.
2. It is possible to compute the set of attractors, or steady states, of the network under analysis.
3. It is possible to compute the basins of attraction of each of the attractors, and characterize them using a special purpose language.
4. A given property can be verified not only on the set of reachable states but also on the sets of states that constitute the attractors and/or the basins of attraction.

We now briefly describe how the new functionalities were created and included in the BioNuSMV tool.

Computation of the Fraction of States that Satisfy a Property

Standard model checkers output a counter-example when a given property is not verified by the system. Since BioNuSMV is based on implicit enumeration of the state transition graph, and the computation of every property is done by manipulating a BDD representation of sets of states, it is possible to compute efficiently the number of states that satisfy (or do not satisfy) a given property. This computation is performed by computing the total number of distinct assignments to state variables that make a given function, represented by a BDD, evaluate to 1. This number is obtained by computing the number of distinct paths that lead to the constant node one, in the BDD, using a dynamic programming technique.

Computation of Steady States

We define a steady state as a terminal strongly connected component (SCC) of the state transition graph. The computation of the SCCs of a graph is a classic problem in computer science. However, the most common solutions have been designed for cases where the graph is stored explicitly. In BioNuSMV the state transition graph is represented implicitly by the transition relation, using a BDD.

BioNuSMV computes steady states using the Lockstep algorithm [Bloem et al., 2006]. The main improvement of this algorithm over the trivial algorithm is that, for each SCC, only one of the two

reachable state sets (backward or forward) is computed. The total number of image computations is of the order of $n\log n$, where n is the number of states in the graph.

Computation of the Basins of Attraction

Since the system is possibly non-deterministic, we must define carefully *basin of attraction* (BOA). In the case of deterministic systems, the basin of attraction of a steady state S is the set of all states x for which there is a path in the state transition graph that ends in one of the elements of S. In the case of non-deterministic systems, there are two possible definitions:

- *weak* basin of attraction - set of states $x \in R$ for which there is *at least one path* from x to one of elements in S.
- *strong* basin of attraction - set of states $x \in R$ for which *all* paths starting in R reach a state in S. This is equivalent to the definition for the deterministic case.

The computation of the basin of attraction of a given steady state can be done using the set of states S that constitute a steady state. The strong basin of attraction of a steady state S is given by the set of states that satisfy the CTL formula AFp where p is a proposition that is true **only** for the states in S. Similarly, the weak basin of attraction of a steady state S is given by the set of states that satisfy EFp.

In our running example there are two steady states, common to the synchronous and asynchronous versions: *(false,false)* and *(true,true)*. For the first steady state, its synchronous and strong asynchronous basins of attraction are the same: the set constituted only by itself; its asynchronous weak basin of attraction has two elements *(false,false)* and *(true,false)*. For the second steady state, the synchronous basin of attraction is equal to the asynchronous weak basin of attraction: {*(false,true), (true,false), (true,true)*}; the asynchronous strong basin of attraction is equal to {*(false,true), (true,true)*}. In the asynchronous version, the strong basin of attraction of (true, true) satisfies the formula, and the weak basin of attraction of *(false, false)* satisfies the formula $\neg x_2$.

Verification of Properties in Attractors and Basins of Attraction

Since BioNuSMV has the ability to compute, using the methods described above, descriptions of the sets of states that constitute an attractor, or a basin of attraction, it is possible to verify if a given property is verified by one of these sets. The implementation of this functionality gives the user a way to specify that the given property should be verified not on the set of reachable states (as is standard in model checkers) but on a specific set of states, obtained in a previous operation.

EXAMPLE APPLICATIONS AND RESULTS

In order to test the effectiveness and usefulness of the improved model checking techniques developed, we applied them to the study of two previously published qualitative models.

Yeast Cell Cycle Network

In the budding yeast (*Saccharomyces cerevisiae*), the periodic alteration of gene transcription levels is the major driving force of the cell-cycle with about 800 genes involved in this process [Cho et al., 1998, Spellman et al., 1998].

The cell-cycle process consists of four phases: G1, growth and preparation of the chromosomes for replication; S, synthesis of DNA and duplication of the centrosome; G2, preparation for; M, mitosis. When a cell is in any phase of the cycle other than mitosis, it is said to be in interphase. Based on literature studies, Li et al. [Li et al., 2004] have proposed and analyzed a Boolean model of the gene regulatory network (Figure 6) that controls the cell-cycle in this yeast. Three classes of regulatory molecules have been considered: cyclins (Cln1, -2, and -3 and Clb1, -2, -5, and -6); the inhibitors, degraders, and competitors of the cyclin complexes (Sic1, Cdh1, Cdc20 and Cdc14); and transcription factors (SBF, MBF, Mcm1/SFF and Swi5).

In this proposed network, checkpoints like the cell size, the DNA replication and damage, and the spindle assembly have also been considered. However, except for the cell size checkpoint, all the other checkpoints were considered always transparent, and they will let the system evolve when necessary. The cell size checkpoint will act as a START signal. The authors studied the attractors of the network dynamics by starting from each of the states of the transition system. They have reported seven steady states, one of them very big with about 86% of the states converging to it. This super stable state corresponds to the G1 stationary state, with genes Sic1 and Cdh1 active. Based on this result the authors concluded that the regulatory network of the Yeast cell-cycle is robust to perturbations.

In this work the model is analyzed assuming a synchronous update mechanism, i.e., that, at any given state, all genes change state synchronously. This assumption, however, overly simplifies the actual dynamics of the network, since different genes will change state at different speeds. We performed an analysis of this network using the same transition function, but assuming a more realistic asynchronous update of the state variables, that considers that different orderings for gene update are possible. We obtained the same seven steady states but the basins of attraction are now of significantly different sizes (see table 1). This can be viewed as an indication that the proposed discrete model is not accurate enough.

Using the functionality made available by BioNuSMV, we have characterized the three largest strong basins of attraction in terms of the active and non-active genes. The most interesting results are the following:

Table 1. Basins of attraction for the 7 steady states of the yeast cell-cycle model, when the state is updated asynchronously

Active genes	Size of weak BOA .	Size of strong BOA
Sic1, Cdh1	1960	14
SBF, Cln1,2	1468	56
MBF, Sic1, Cdh1	1074	12
Cdh1	1737	1
No active genes	1760	2
Sic1	1935	2
MBF, Sic1	1036	2

- The cyclin Cln3 is off in all states that belong to a strong basin of attraction. This means that as long as Cln3 is active the system cannot commit to one of the steady states.
- In the strong basin of attraction of the (Sic1, Cdh1) steady state transcription factors SBF and MBF are always off.
- In the strong basin of attraction of the (SBF, Cln1,2) steady state, transcription factor SBF is always on and transcription factor MBF is always off.
- In the strong basin of attraction of the (MBF, Sic1, Cdh1) steady state, transcription factor SBF is always off and transcription factor MBF is always on.

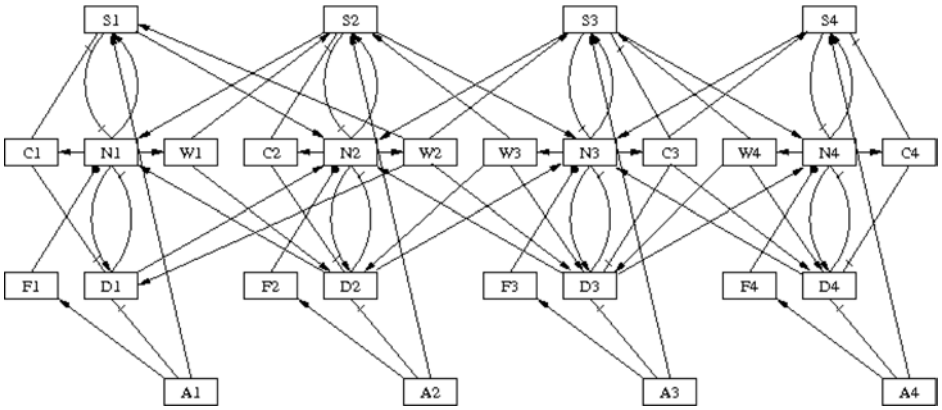
The last three points suggest that the ability to turn off simultaneously the activity of the transcription factors SBF and MBF plays an important role in determining the steady state of this network. This hypothesis is corroborated by the detailed continuous model of Chen et al. [Chen et al., 2004].

Formation of the Dorsal-Ventral Boundary in *D. Melanogaster* Wing Imaginal Disc

The second example we explored is a four cell discrete model of the gene regulatory network that controls the formation of the dorsal-ventral boundary in the wing imaginal disc of the fruit fly [Gonzalez et al., 2006a].

The proposed modeling approach followed the generalized logical formalism previously developed by R. Thomas et al. [Thomas et al., 1995]. In this model seven regulatory elements were considered: Apterous (Ap), Cut (Ct), Delta (Dl), Fringe (Fng), Notch (N), Serrate (Ser) and Wingless (Wg). The set of values associated with these regulatory elements is {0, 1} for all the elements, excepting DI and Ser. In the case of these two regulators they can take the values 0, 1 or 2, corresponding to negligible, low and high product levels. Another aspect is that the framework proposed is a multi-cellular framework. Four cells were considered (Figure 7) to model the most crucial inter-cellular interactions involved in the expression of *wg* (*wingless*) at the dorsal-ventral (DV) boundary. Two boundary cells were used to model

Figure 7. Graph model for the regulatory network that controls the creation of the dorsal-ventral boundary in D. melanogaster [Gonzalez et al., 2006a]. Positive interactions are denoted by normal arrows, negative interactions by blunt arrows and ambivalent interactions by bullet arrows.



the interactions across the DV boundary during the early signaling process. To model the interactions between boundary cells and their adjacent cells, two additional cells were included. These interactions appear in the mid-third larval instar.

In this work the authors used a fully asynchronous model and computed the steady states of the model using constraint programming. They obtained five singular steady states, only one of them with a genetic profile compatible with the existence of the boundary in the two middle cells. This somewhat negative result led us to investigate further the characteristics of this network.

After replicating the results of Gonzalez et al., [Gonzalez et al., 2006a], we computed and characterized the basins of attraction in order to gain a better understanding of the dynamics of the system.

One important remark is that the strong basin of attraction of the biologically observed steady stated is smaller than two of the other ones. In fact, only very particular orderings of changes in state variables led to the convergence to the desired steady state. We once again checked for the presence and absence of active genes in the strong basins of attraction.

This analysis suggested that we should somehow trim the space of reachable states by restricting some of the possible orderings of change of state variables to coerce the model to more accurately reflect the biological observations. One such restriction that is sensible from the biological point of view is to allow for a gene to be turned off only after it has activated its targets. We have imposed this restriction on Notch. This corresponds to assuming that sustained production of Notch in a cell always leads to the production of Cut and Wingless on the same cell. This modification led to an increase in the size of the strong basin of attraction of the real steady state by a factor of one hundred. However, this restriction was not enough to make the remainder four steady states disappear. It was still possible in this version of the model for a cell in a boundary to see the activity of Notch down-regulated and an increase in the activity of Delta and Serrate, leading to the appearance of a cell with boundary characteristics in one of the flanking cells.

In [Chen et al., 2004] a continuous model for the same system is presented. The authors describe similar difficulties in obtaining a robust model. To solve this problem they introduce a new hypothesis, that boundary cells are not sensitive to the effect of the Wingless gene. This means that Delta and Serrate no longer can be up-regulated in the boundary cells, preventing the loss of Notch activity in these cells. After we introduced this modification in the discrete model, we observed that the system has only one steady state, the one observed *in vivo*.

This result shows that the ability to perform the type of analysis made available by BioNuSMV can be instrumental in our quest for good experimental hypotheses that help elucidate the dynamic behavior of gene regulatory networks.

Table 2. Sizes of the basins of attraction of the regulatory network that controls the formation of the dorsal-ventral boundary in the wing imaginal disc of the fruit fly

Active genes	Size of weak BOA	Size of strong BOA
–	241598788	5440
Dl,Ser / N,Ct,Wg / N,Ct,Wg / Dl,Ser	242620820	11424
N,Ct,Wg / Dl,Ser / Dl,Ser / N,Ct,Wg	243710054	21384
Dl,Ser / N,Ct,Wg / Dl,Ser / N,Ct,Wg	246633046	2496032
N,Ct,Wg / Dl,Ser / N,Ct,Wg / Dl,Ser	245750924	1795136

CONCLUSION AND FUTURE WORK

We have presented a methodology and a tool for the analysis of gene regulatory networks that enables researchers to analyze and characterize the state space of the system under analysis.

The ability to compute and characterize the basins of attraction of the computed steady states is useful, and can be used to improve the models, by helping in the identification of the reasons for lack of robustness and by suggesting corrections and refinements. In the two cases under study, we were able to propose changes in the model that led to more accurate predictions.

In more general terms, we argue that discrete models coupled with model checking techniques can be used effectively to grasp important characteristics of biological systems. A tool that supports these analyzes, BioNuSMV, a model checker with extended functionality, has been developed and is available to the research community.

The software described in this chapter is publicly available, and can be obtained by contacting the authors of this article.

REFERENCES

Biere, A., Cimatti, A., Clarke, E., & Zhu, Y. (1999). *Symbolic model checking without BDDs*. Springer.

Bloem, R., Gabow, H. N., & Somenzi, F. (2006). An algorithm for strongly connected component analysis in n log n symbolic steps. *Formal Methods in System Design*, *28*(1), 37–56. doi:10.1007/s10703-006-4341-z

Bryant, R. E. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, *C-35*(8). doi:10.1109/TC.1986.1676819

Chabrier, N., & Fages, F. (2003). *Symbolic model checking of biochemical networks*. (. *LNCS*, *2602*, 149–162.

Chen, K. C., Calzone, L., Csikasz-Nagy, A., Cross, F., Novak, B., & Tyson, J. J. (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, *15*(8), 3841–3862. doi:10.1091/mbc.E03-11-0794

mCho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., & Wodicka, L. (1998). A genomewide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, *2*(1), 65–73. doi:10.1016/S1097-2765(00)80114-8

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., et al. (2002). NuSMV version 2: An opensource tool for symbolic model checking. In *Proc. International Conference on Computer-Aided Verification (CAV 2002)* (Vol. 2404 of LNCS), Copenhagen, Denmark. Springer.

Clarke, E. M., & Emerson, E. A. (1981). Characterizing properties of parallel programs as fixpoints. In *Seventh International Colloqium on Automata, Languages, and Programming* (Vol. 85 of LNCS).

de Jong, H., Geiselmann, J., Hernandez, E., & Page, M. (2003). Genetic network analyzer: Qualitative simulation of genetic regulatory networks. [Evaluation Studies.]. *Bioinformatics (Oxford, England), 19*(3), 336–344. doi:10.1093/bioinformatics/btf851

de Micheli, G. (1994). Synthesis and optimization of digital circuits. McGraw-Hill.

Fages, F., Soliman, S., & Chabrier-Rivier, N. (2004). Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *J Biological Physics and Chemistry*, *4*(2), 64–73. doi:10.4024/2040402.jbpc.04.02

Garg, A., Xenarios, I., Mendonza, L., & DeMicheli, G. (2007). An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments.) . *LNCS*, *4453*, 62–76.

Gonzalez, A., Chaouiya, C., & Thieffry, D. (2006a). Dynamical analysis of the regulatory network defining the dorsal-ventral boundary of the Drosophila wing imaginal disc. *Genetics*, *174*(3), 1625–1634. doi:10.1534/genetics.106.061218

Gonzalez, A., Naldi, A., Sanchez, L., Thieffry, D., & Chaouiya, C. (2006b). GINsim: A software suite for the qualitative modelling, simulation, and analysis of regulatory networks. *Bio Systems*, *84*(2), 91–100. doi:10.1016/j.biosystems.2005.10.003

Hasty, J., McMillen, D., Isaacs, F., & Collins, J. (2001). Computational studies of gene regulatory networks: In numero molecular biology. *Nature Reviews. Genetics*, *2*(4), 268–279. doi:10.1038/35066056

Li, F., Long, T., Lu, Y., Ouyang, Q., & Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(14), 4781–4786. doi:10.1073/pnas.0305937101

McMillan, K. L. (1993). *Symbolic model checking*. Kluwer.

McMillan, K. L. (1999). *The SMV system*. Cadence Berkeley Labs.

McMillan, K. L. (2003). Interpolation and SAT-based model checking. In CAV (Vol. 2725 of LNCS, pp. 1–13). Springer.

Shults, B., & Kuipers, B. (1997). Proving properties of continuous systems: Qualitative simulation and temporal logic. *Artificial Intelligence*, *92*(1-2), 91–129. doi:10.1016/S0004-3702(96)00050-1

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., & Eisen, M. B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae microarray hybridization. *Molecular Biology of the Cell*, *9*, 3273–3297.

Thomas, R., Thieffry, D., & Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks - I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, *57*(2), 247–276.

KEY TERMS AND DEFINITIONS

Cell Cycle: Is the series of events that take place in a cell leading to its division and duplication (replication).

Computation Tree Logic: It is a branching-time logic, meaning that its model of time is a tree-like structure in which the future is not determined; there are different paths in the future, any one of which might be an actual path that is realized.

Dorsal Ventral Boundary: Boundary between different cell types at the Drosophila wing.

Gene Regulation: The processes that cells and viruses use to turn the information in genes into gene products.

Gene Regulatory Networks: A gene regulatory network is a collection of DNA segments in a cell which interact with each other and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA.

Model Checking: To algorithmically check whether a simplified model of a system satisfies a given specification.

State Transition Graph: A graph consisting of circles to represent states and directed line segments to represent transitions between the states.

Chapter 17
Determining the Properties of Gene Regulatory Networks from Expression Data

Larry S. Liebovitch
Florida Atlantic University, USA

Lina A. Shehadeh
University of Miami, USA

Viktor K. Jirsa
Florida Atlantic University, USA

Marc-Thorsten Hütt
Jacobs University, Germany

Carsten Marr
Helmholtz Zentrum München, Germany

ABSTRACT

The expression of genes depends on the physical structure of DNA, how the function of DNA is regulated by the transcription factors expressed by other genes, RNA regulation, such as that through RNA interference, and protein signals mediated by protein-protein interaction networks. We illustrate different approaches to determining information about the network of gene regulation from experimental data. First, we show that we can use statistical information of the mRNA expression values to determine the global topological properties of the gene regulatory network. Second, we show that analyzing the changes in expression due to mutations or different environmental conditions can give us information on the relative importance of the different mechanisms involved in gene regulation.

DOI: 10.4018/978-1-60566-685-3.ch017

INTRODUCTION

All living things contain a "memory" of the past that explicitly defines their species and implicitly reflects the evolutionary events that led to their species. Typically, this memory is encoded in deoxyribonucleic acid, DNA, although it may also be encoded in ribonucleic acid, RNA (for retroviruses), or in epigenetic coding (such as methylation of DNA), or in three dimensional structures (such as the protein confirmations of prions). Each organism uses this memory as a blueprint to design and maintain itself. But it is not like a blueprint that we use to build buildings which is a smaller symbolic picture of the building. Rather, it is more like a computer code, which when executed generates structures that have a very different form than the code itself. But it is unlike the computer codes that we currently construct. Our computer codes execute their instructions in a preset order. However, which instructions living things execute are chosen by a multilevel cacophony of highly interacting networks.

The Central Dogma of molecular biology (Crick, 1958) was that genetic expression is a one way street from the transcription of DNA into mRNA, and then the translation of mRNA into protein. But we are now beginning to appreciate that multiple processes, both forward and backward, control and edit how the instructions of DNA are executed into the proteins that form the structure and function of cells. In this chapter we explore how networks control DNA expression, from within DNA (depending on the physical structure of DNA and the regulation that one gene exerts on another), and from outside of DNA (depending on the editing of mRNA and protein regulatory networks). We show how understanding the physics of networks can be used to devise methods of analysis that reveal the global and local organization of these networks.

BACKGROUND

Transcription Regulatory Networks (TRN)

In transcriptional regulation, the product of one gene, a transcription factor (TF) protein, binds to the promoter region of another gene and increases or decreases its expression. The discovery of regulatory processes in the *lac* operon (Jacob & Monod, 1961) marked an historic step in biology. Lately, the assembly of many such effects into a full network of genetic interactions has heralded the emergence of a system-wide view on transcriptional regulation (Thieffry et al., 1998): The transcriptional regulatory network (TRN) describes how genes regulate each other through the expression and binding of their TFs. In mathematical terms, the TRN is a directed graph consisting of nodes representing genes and links representing the directed regulatory interaction between two genes, mediated by a TF. The statistics of the topology of these connections are summarized by the in-degree and out-degree distributions which define the number of genes with a given number of incoming and outgoing connections.

In the bacterium *Escherichia coli*, evidence for the regulatory action of TFs is documented in what is ''currently the largest electronically-encoded database of the regulatory network of any free-living organism'' (Salgado et al., 2006), called RegulonDB. The most recent version (5.6) of the publicly available database comprises 2735 interactions between 1345 genes. A small fraction of genes (79) are top-level regulators with no input from other genes, while 1197 nodes are solely target nodes, with no regulatory output to other genes. Various studies used the information contained in RegulonDB to construct the *E. coli* TRN and, e.g., analyzed its motif content (Shen-Orr et al., 2002), the aggregation of such motifs

into larger clusters (Dobrin et al., 2004), and global topological features, as its modular and hierarchical structure (Ma et al., 2004). Also for another model organism, the yeast *Saccharomyces cerevisiae*, large TRNs have been constructed and investigated, either by single high-throughput experiments (the ChIP on Chip approach of Lee et al. (2002, 2007) revealed over 2000 targets of 106 yeast TFs) or by compiling the knowledge from different databases into one large network (Guelzim et al., 2002). A comparison of the TRNs of *E. coli* and yeast revealed a common hierarchical structure and surprising functional aspects (Yu & Gerstein, 2006): Most essential TFs, i.e. those proteins for which a gene knock-out leads to no longer viable cells strains, are in between the top-level regulators and the target genes.

Apart from using and analyzing databases, so called 'reverse engineering' approaches try to infer the topology of TRNs from gene expression profiles with a diverse toolbox. The emerging networks have been used to predict interactions in the *E. coli* TRN (Gardner et al., 2003) or propose a modular design of the yeast TRN (Ihmels et al., 2002). These techniques become especially relevant when applied to the expression profiles of human tissues (see, e.g., Basso et al. (2005) for an application to human B cells) or a species with mostly unknown regulatory interactions.

An interesting topological feature of the TRN is that it has an almost treelike structure (particularly for *E. coli*, but also for yeast, *S. cerevisiae*; see Alon 2006 for a detailed discussion). This observation has several consequences. First, hierarchical levels in the network can be meaningfully analyzed (Yu and Gerstein 2006). Secondly, it leads to the question, how information can be circulated in the network, when there is a dominant directed flow in the network dictated by its architecture. It becomes ever more transparent that additional interactions beyond transcription factors, particularly regulation based on protein interactions but also based on small regulatory RNA, disrupt this general feedforward structure and relay signals from the bottom layer again to the top-level input nodes (Yu and Gerstein 2006; Shimoni et al. 2007, Tsang et al. 2007).

The Network that Regulates Gene Expression is More than Just the TRN

It is highly important to note that the TRN is only one component of the gene regulatory network. Transcription of genes into RNA is in large measure regulated by proteins that bind DNA. These interactions include the basal transcription apparatus recognizing the core promoter and its associated general transcription factors (reviewed in Arnone and Davidson, 1997); a host of other more specialized transcription factors that combinatorially regulate the transcription of specific subsets of genes through DNA-binding events (Thanos and Maniatis, 1995); nucleosome-forming histones that regulate the structure of chromatin fibers; and factors like histone acetylases that make the DNA more accessible to transcription factors (reviewed in Roth et al 2001) or DNA methylases that mark the genome epigenetically, in some cases completely silencing gene expression (reviewed in Jaenisch and Bird, 2003; Levine and Tjian, 2003; Wray *et al.*, 2003).

Transcribed non-coding RNAs, like micro RNAs and RNA binding proteins, also play a significant role in gene regulation by affecting mRNA stability and degradation. In addition to regulation of mRNA levels, regulatory complexity and protein diversity are further controlled at the level of splicing, which involves many RNA binding proteins that act in a combinatorial manner (reviewed in Mata *et al.* 2005). This level of post-transcriptional regulation can be important but is usually overlooked in interpreting, for example, microarray data. Most technologies measure the steady-state levels of mRNA, which are the result of both the rate of transcription and of RNA turnover. Thus, different events and pathways are involved in stabilizing the transcript, in eliminating faulty transcripts (Fasken and Corbett, 2005) and

in adjusting the level of mRNA to the physiological needs of the organism (Khodursky and Bernstein, 2003). Many of these regulatory events involve the presence of specific sequences in the 3'UTR, which are bound by different ribosomal-binding proteins (Parker and Song, 2004). There is now an opportunity to integrate functional and structural data, to understand how the biophysical aspects of protein-nucleic acid interactions affect their functions. Until recently protein-nucleic acid interactions have largely been studied either through structural approaches (for example, structure determination of the polymerase holoenzyme) or through computational approaches based on sequence data (for example, de novo motif finding algorithms, or phylogenetic footprinting) or functional expression data (for example, using gene expression data to infer genetic regulatory networks). Recent technological advances have enabled many different types of data to be gathered at a genome-wide and proteome-wide scale, including: genomic sequences, tissue-specific ESTs and mRNA expression data, the abundance of various RNA populations, protein-protein and protein-ligand interactions, chromosomal interactions, and protein-nucleic acid binding data. In addition, efforts in structural biology are yielding structural data on proteins, protein complexes, and protein-ligand interactions.

Taken all together these multiple levels of interactions define the phenomenological model of how the activity of genes affects the activity of other genes (as well described, for example, by Brazhnik et al. 2002). In this chapter we use the phrase "gene regulatory network" to refer to the combined effects of all these interactions. We will use more restricted terms, such as "transcription regulatory network (TRN)" when we refer to only one component of the gene regulatory network, in this case those interactions regulated by transcription factor proteins that bind to the regulatory regions of genes and alter their expression.

The new data from all these different experimental approaches provide new opportunities for an integrative approach. Combining the previously distinct perspectives of structural, functional, and genomic analyses should improve our ability to identify essential biological associations, and ultimately to model and predict these interactions. Studies combining these different types of analysis and data spurs new collaborations between researchers in these historically distinct fields.

NETWORKS

Types of Networks

The statistical and graph theoretical analysis of networks has become a rapidly evolving field in recent years, not least due to the discovery of consistent structural principles of "small world networks" including clustering (Watts and Strogatz, 1998) and scale-free attributes (Albert and Barabasi, 2002). Numerous studies suggest that most scientifically and technologically significant large-scale networks ranging from social networks to cellular metabolism and the internet are neither random nor regular, but instead share common principles of organization (Strogatz, 2001). For the analysis of networks, several graph theoretical measures, such as the network's clustering index and its characteristic path length, are of particular interest (described in more detail below). The cluster index captures the extent to which a unit's neighbors connect to each other, forming a "clique" or local cluster. The characteristic path length is the average length of the shortest directed path between any two units in the network. The shorter the characteristic path length, the "closer" (in terms of distance in graphs), on average, are the network's units.

Random networks have been studied by Erdös & Rényi (1959) using random-graph theory. A graph is a pair of sets {*P,E*} where *P* is a set of *N* nodes and *E* is a set of edges connecting two elements of *P*. The graph is typically illustrated by dots corresponding to nodes and by lines corresponding to edges. Every pair of nodes is connected with equal probability *p* and the majority of nodes have approximately the same degree, close to the average degree ξ of the network. The degree distribution of the random network is a Poisson distribution with a peak at $P(\xi)$. Random-graph theory studies the properties of the connection probability associated with graphs with *N* nodes as $N \rightarrow \infty$.

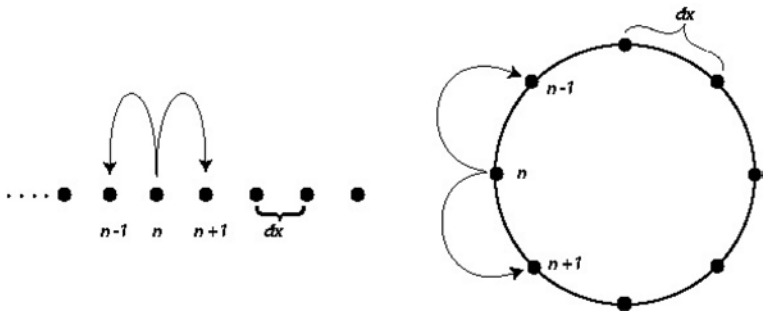
Scale-free networks are characterized by a hierarchy of connections that is self-similar across different scales or levels of structure and thus obeys a power law degree distribution of the form $P(k) = Ak^{-a}$ (Barabasi & Albert 1999). In a scale free network, structures extend over a wide range of scales. Such network topologies develop when new connections are added preferentially to nodes that already have many connections (Huberman et al. 1998; Adamic & Huberman 1999; Albert & Barabasi 2002).

Small world networks are characterized by a short characteristic path length and a high clustering coefficient (Albert & Barabasi 2002). Given the same number of nodes *N* and the average number of connections ξ, both a random network and a small world network have a similarly small average path length *l*. Under these same conditions a small world network has a higher clustering coefficient than a random network.

Analyzing the Connectivity of Networks

The effective geometry in which the dynamics of a system evolves is determined by its connectivity matrix, together with the boundary conditions of the system. A symmetric connectivity matrix w_{ij} is given when its elements satisfy the condition $w_{i,j} = w_{j,i}$. The stronger constraint of translational invariance requires that the values of the matrix elements are a function of the difference *i-j* only rather than a function of the absolute value of the indices *i,j*. This formulation is mathematically precise. A slightly weaker formulation for near-neighbor connections is given in the following. An effective one-dimensional geometry will be achieved, if the network nodes *i* may be indexed such that elements in the connectivity matrix w_{ij} may be ordered "around" its diagonal within a width which approximately remains the same along the diagonal. A meaningful choice of an index scheme will identify the neighborhood as shown in Figure 1. As long as most of the elements of the connectivity matrix follow this distribution scheme,

Figure 1. One-dimensional connectivity with open boundary conditions (left) and periodic boundary conditions (right)



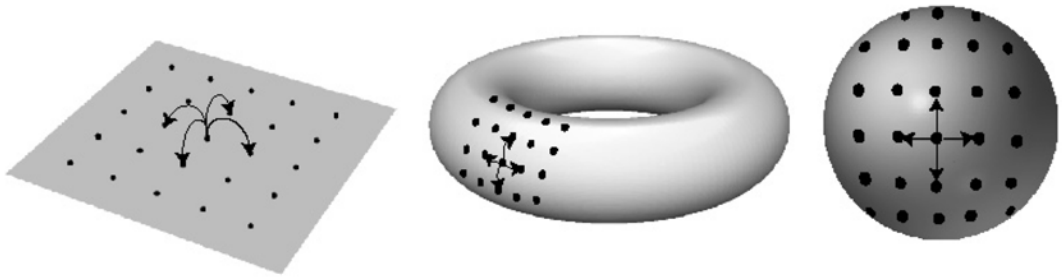
the effective physical geometry will be one-dimensional. The boundary conditions will determine if the one-dimensional space, the line, will be closed for periodic boundaries or open otherwise.

An effective two-dimensional geometry will be obtained if an indexing scheme is adopted in which the neighborhood relations of the connectivity matrix may be expressed as shown in Figure 2. Again, as long as most of the elements of the connectivity matrix are ordered around its diagonal within an approximately constant width for each space dimension, then the effective geometry will be a two-dimensional surface. If the boundary conditions are periodic, then closed surfaces are obtained which can take either the form of a two-dimensional torus or a sphere or ellipsoid. In principle, it is possible to construct many two-dimensional closed surfaces with arbitrary topologies, but these three, torus, sphere and ellipsoid, are the most common in applications. Open two-dimensional surfaces typically obey zero amplitude or zero flux boundary conditions.

Other Measures that Characterize Networks: Path Length and Clustering Coefficient

Network topologies are characterized from the perspective of statistical mechanics by the number of nodes N and by the number k of connections (or ‘’edges’’) to other nodes. Since not all nodes in the network have the same number of connections (referred to as the node degree), the spread in the node degrees is characterized by a distribution function P(k), which gives the probability that a randomly selected node has k connections. Among many measures to characterize network topologies, two well known measures are the average path length and the clustering coefficient. The average path length *l* of a network is defined as the number of edges in the shortest path between two nodes, averaged over all pairs of nodes (Watts and Strogatz, 1998; Albert and Barabasi, 2002). The clustering coefficient C_i of node *i* that has k_i edges which connect it to k_i other nodes, is defined as $Ci = \frac{2E_i}{k_i(k_i - 1)}$ where E_i is the number of edges that actually exist between the k_i nodes. The total number of all possible edges is $k_i(k_i\text{-}1)/2$. The clustering coefficient of the whole network is the average of all the C_i from every node *i*. The in-degree and out-degree are computed as the number of incoming and outgoing connections to/from a node, respectively. The degree is the sum of in- and out-degree. The betweeness centrality is the fraction of shortest path between any two pairs of nodes passing through a particular node. If *s* is the source and *t* is the target node, then n_{st} is the total number of shortest paths linking these and n_{st} *(i)*

Figure 2. Two-dimensional connectivity with open boundaries (left), double periodic boundaries (torus, middle) and spherical boundaries (right)



passes through node *i*, Then betweeness centrality of the *i*-th node is calculated as $\frac{1}{N(N-1)}\sum_{s,t}\frac{n_{st}(i)}{n_{st}}$. Albert and Barabasi (2002) wrote a detailed review on the statistical description of network topologies and Sporns & Tononi (2007) provide a detailed description of network connectivity measures.

Relationship Between the Structure and the Dynamics of a Network

In the last few years, a wide range of disciplines have intensely studied how the topology of a network shapes, regulates, or even enhances dynamic processes on the network. For gene regulatory networks this question is particularly interesting, because an observed expression pattern may be viewed as the result of the dynamics of the interacting genes in the gene regulatory networks. Very obviously many other biological mechanisms contribute to gene expression levels in similar proportions (like DNA topology and the regulatory action of microRNA or protein-protein interactions not incorporated in the gene regulatory network based on transcription factors and their binding sites, just to name a few). An important future task is thus to isolate and understand the systematic contribution of dynamics on the gene regulatory network to gene expression levels. Here we briefly summarize a few general results on the link between topology and dynamics, which may be useful for understanding the potential impact of network topology on expression levels. Inspite of the very interesting findings on the relation of topology and dynamics, e.g., for phase oscillators (Arenas et al. 2006), synchronization in general (Nishikawa and Motter 2006; Atay and Biyikoglu 2004), epidemics (Moreno et al. 2002) and excitable dynamics (Graham and Matthai 2003; Roxin et al. 2004; Müller-Linow et al. 2006, 2008) or chaotic oscillators (Yook and Meyer-Ortmanns 2006), we will here focus on binary dynamics, as they may prove most helpful for the study of gene regulation.

A simple and very successful mathematical model of gene regulation has been formulated several decades ago by Kauffman (1969). It describes the interaction of binary elements in a random graph: In a network consisting of nodes (“genes”), every single node is regulated by other randomly selected nodes via definite Boolean functions (random Boolean network). The pattern of transitions between system states, the attractor structure of the system, can be thought of as a highly simplified model of cell differentiation: different attractors correspond to different cell types; a basin of attraction in this general scheme corresponds to the range of initial conditions (and, in a sense, of environmental cues) leading to this attractor (or cell type). Based on the general framework of random Boolean networks and their extension to general network topologies (and particularly the use of threshold dynamics), huge progress has been made in the last few years in linking observed properties of the dynamics with topological features of the graph (see, e.g., Bornholdt, 2005).

Another modeling approach for complex biological systems are cellular automata (CA), which are Markovian dynamics on a finite state space. Proposed by von Neumann (see von Neumann 2001) as a model system for biological self-reproduction, a surge of research activity from the 1980’s onwards (Wolfram 1983) established them as a standard tool of complex systems theory. Cellular automata on graphs in principle allow assessing dynamical changes due to variation of graph topology (Marr and Hütt 2005).

The appearance of network motifs (i.e. patterns of interconnections that occur in a network far more often than expected at random) in transcriptional regulatory networks can be motivated using simple dynamical models (see Alon 2007 for an overview). There is also a similarity in motif content of func-

tionally similar networks (Milo et al. 2004), where motifs are groups of few nodes with a specific link pattern. The subnetwork frequencies in genetic networks (Milo et al. 2004) have been shown (Klemm and Bornholdt 2005) to correlate with the dynamic robustness profile of these subnetworks: frequent three-node subnets (compared to a randomized graph) have the highest robustness of Boolean dynamics under a noisy update scheme.

Another interesting issue in the context of network dynamics is noise, which is known to be an important factor in quantitatively understanding gene regulation (McAdams and Arkin 1997). At the same time, noise has turned out to affect network dynamics in sometimes counterintuitive ways, being compensated (Moreira et al. 2004) or mimicked (Graham and Matthai 2003; Marr and Hütt 2006) or distorted (Amaral et al. 2004) by the network topology.

Spectral Properties of Network Matrix Representation

We have seen above that a graph can be represented in terms of its connection matrix which is also called its adjacency matrix. This representation allows comparing topological properties of the graph with spectral properties of the matrix. A theoretical foundation of this approach is essentially given by random matrix theory, where it has been shown that the frequency distribution of eigenvalues of many random symmetric matrices has a shape proportional to $\sqrt{R^2 - \lambda^2}$, where R delimits the range of eigenvalues and depends on the details of the random matrix (Wigner's semi-circle law; Wigner 1958). Networks can be classified according to deviations from this eigenvalue distribution.

Spectral properties of the adjacency matrix or, alternatively, the graph Laplacian (which is the diagonal matrix of the degree sequence minus the adjacency matrix) have been studied. Key results include: attempts to classify networks according to deviations from this eigenvalue distribution (Banerjee and Jost 2007) and the observation that systematic gaps in the ranked eigenvalue sequence (as found, e.g. in hierarchical networks) organize the route towards synchronization of phase oscillators on a graph (Arenas et al. 2006).

Relations between topology and dynamics are produced by the interplay between eigenvalue sizes and the associated eigenvectors. The eigenvectors essentially are directions in node space, along which dynamic processes under certain conditions organize. At this intersection point of graph topology and linear stability analysis known from dynamical systems theory, particularly the eigenvector to the largest eigenvalue offers insight into the asymptotic behavior.

Let us return to the view of expression as a dynamic process on a network. The linear spread of excitations in the network is a simple realization of dynamics, which can at the same time be formalized to be an iterative prescription for computing the eigenvector of the graph belonging to its largest eigenvalue. Recently, the components of the eigenvector of the adjacency matrix have been used to model absolute cDNA microarray expression levels (Shehadeh et al. 2006). As an appropriate surrogate on directed networks, one can weight the nodes with respect to whether these nodes are hubs (connected to many other nodes) or authorities (connected by many other nodes), see Kleinberg 1999. Those distributions have proved useful in identifying functionally important nodes in networks such as social networks or power grids or in identifying the best targets for internet search engines (as those linked to the authorities). A linear relationship between expression and the importance of a node as a hub would imply an adaptation to the out-degree and therefore an information propagation reverse to the direction of the regulatory interaction. As soon as more data become available, particularly a wider range of TRNs, it

will be informative to search for resemblances of such patterns with the corresponding features of expression profiles. Eventually, a view may emerge, that the topological constraints the TRN imposes on gene expression resemble the iterative process of computing the TRN's Perron-Frobenius eigenvector.

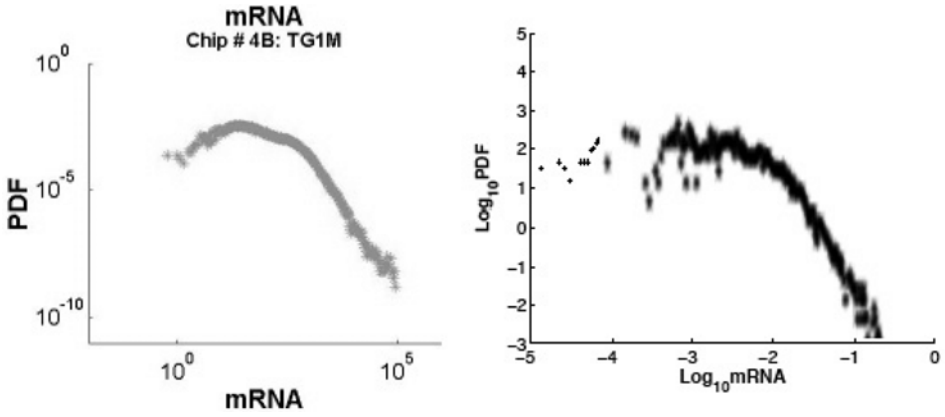
Global Expression Properties

These conceptual properties of the networks described above can be used to characterize the nature of gene regulatory networks in a new way. Previously, analysis has focused on a bottom–up approach, trying to piece together all the separate interactions between individual genes from the experimental data. The concepts from network analysis now make possible a top–down approach, trying to discover the overall organization of this network from experimental data. This approach (Shehadeh et al 2006; Liebovitch et al 2006) uses the fact that different network structures will produce different statistical patterns in the mRNA expression. Thus, the statistics of the mRNA expression can be used to gain valuable information on the gene regulatory network. For example, the structure defined by the connectivity matrix w_{ij} can be related to the dynamics, the mRNA expression levels X^n at time n, by iterating $X^{n+1} = WX^n$ until the expression levels reach a steady state. We formulated some basic models of different types of gene regulaion networks (both random and scale-free), computed the statistical properties of the mRNA levels produced by those models which could then be compared to the statistical properties of experimentally data. Moreover, we have found that there is an important relationship between the structure of the network as described by the degree distribution, P(k), the probability that a node is connected to k other nodes, and the dynamics of the network as described by the probability density function, PDF, given by f(x), the probability that the mRNA expression is in the range (x,x+dx). For linear networks f(x) is proportional to P(k), and for nonlinear networks the tail of the distribution of f(x) has the same functional form as P(k). This is an important finding that shows that there is a strong interrelationship between the dynamics and the structure of a network.

Practical Example #1: Determining the Properties of the Gene Regulatory Network in the Fruit Fly *Drosophila melanogaster*

We now show how these network concepts can be applied to the practical analysis of experimental data and what is learned from that analysis. We computed the statistical pattern of mRNA expression as measured by the PDF of mRNA expression levels from different models of random networks having different average numbers of connections, scale free networks having different scaling exponents, networks with either similar or different in-degree distributions and out-degree distributions, and networks having different average path lengths and clustering coefficients. We then compared the mRNA PDF from those models to the mRNA PDF measured from 54 expression sets where cDNA microarray technology was used to measure mRNA expression levels of virtually every gene in the heads of control and period null mutant Drosophila flies (Shehadeh et al 2006; Liebovitch et al 2006). An example of these results is shown in Figure 3. The experimental data was best represented by the PDFs of the scale free models of gene regulatory networks with scaling exponents in the range 1.5 to 2.0. These results were accomplished without knowing which gene is responsible for what mRNA level. This supports the feasibility of our basic idea that we can use the global, statistical information from the observed mRNA levels to infer information about the pattern of genetic interactions. This work still cannot uniquely determine the topology of a genetic network since it lacks the proof that the PDFs are unique for each

Figure 3. PDF of the mRNA expression levels of wild type Drosophila melanogaster (left) compared to a linear network model with symmetric in-degree and out-degree scale free distributions with scaling exponent equal to 2 (right)



model. However, this work suggests that like many other natural networks, genetic networks seem to have scale free topology.

These results may also provide an aid to screen biological systems to determine which ones are the best candidates for therapeutic intervention. A system where the mRNA expression levels tell us that the genetic interactions depend on a balanced interaction of a very large number of genes, such as the models with random topology, may be a poor candidate system for basic scientific studies or clinical applications. There are just too many simultaneous interacting genes so that experiments or therapeutic treatments that alter one gene at a time will not be productive. A system that has a scale-free structure or a system in which there is only a limited number of genetic interactions or a strong hierarchy of connections, such as models with a scale-free topology, may be a good candidate system for basic scientific studies or clinical applications. The small number of simultaneously interacting or controlling genes may make it possible to study the effects of each of these genes separately.

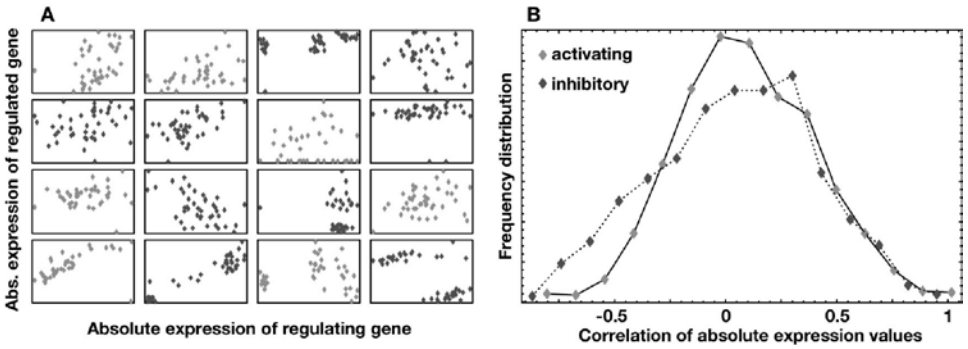
ROLE OF THE TRANSCRIPTION REGULATORY NETWORK

The expression of a gene is a highly complex procedure that involves transcribing a stable DNA sequence into an unstable messenger molecule (mRNA) and then translating the mRNA into a functional protein. The regulation of gene expression is at the core of many important biological processes and a central determinant of how organisms develop. It takes place on very different levels of cellular organization: from changes in the physical structure of DNA (thereby inhibiting or facilitating the access of the RNA polymerase to a gene's promoter), over the specific binding of transcriptions factors (TFs) and their mutual interactions, the natural, or possibly miRNA induced degradation of mRNA to, finally, the regulation of translation at the ribosome. The regulation of transcription through TFs is, however, considered as the most direct route in this ensemble of regulatory actions. It is at the very beginning of mRNA synthesis and thus reliable and cheap. The global interaction pattern of TFs can be collectively described by the transcriptional regulatory network (TRN).

The ability to explicitly measure gene expression in many genes at once opens remarkable possibilities in determining the functional roles of expression patterns (see, e.g. Ideker et al. (2002) for a thorough analysis of the yeast galactose-utilization pathway with several high-throughput methods). In a very simplified view one can think of gene expression observed in microarray experiments as a result of a dynamic process on the TRN. This view is our starting point for studying whether the properties of the underlying TRN can be detected in the gene expression data. In light of the fact that gene expression is determined by many processes (like physical properties of DNA, mRNA interference, the protein-protein interaction network, and the environment), our aim is to find the specific role and the relative importance of the TRN topology. We thus use the most complete TRN available, the TRN of the bacterium *Escherichia coli*. It is electronically encoded in the RegulonDB database (Salgado et al., 2006), which includes the genes' names, positions and operon association, together with known regulatory activating and inhibitory interactions. In what follows, we relate topological properties of genes in the TRN to the genes' expression profiles. In doing so we pass from the scale of individual links (i.e. pairs of nodes) to sub-networks formed by all nodes topologically downstream of a particular node and then, finally, to the large-scale topological properties.

The simplest topological entity of the TRN is a link that connects a pair of genes. We can label each link with its regulatory characteristics as activating or inhibitory, according to its database annotation. To see if there is a consistent clear correlation in expression between the regulating gene and the regulated gene under different environmental and experimental conditions, we analyze the microarray data of *E. coli* aerobic shift experiments (Covert et al., 2004), available from the ASAP database (Glasner et al., 2003). Finding a substantial difference in the correlation values for activating and inhibitory links, respectively, would be the clearest signal for gene expression being dominantly determined by gene-gene interaction. Figure 4A shows scatter plots of the absolute expressions of regulated versus regulating genes for a

Figure 4. Correlation of expression and TRN link characteristics. **A** *Scatter plots of the expressions of the regulating gene versus the regulated gene for 16 arbitrarily selected transcriptional regulatory interactions in the largest connected component of the E. Coli TRN. The expression levels of genes connected by inhibitory interactions are shown in dark gray, activating interactions in light gray. No immediate correlation between expression and the type of regulatory interaction can be observed. This impression does not change if we consider whole operons or only genes without self-regulation.* **B** *The frequency distribution of the Pearson's correlation coefficients for inhibitory (dark gray) and activating (light gray) links, normalized to the area under the curve. Both distributions look very similar, centered around a zero correlation. Only the elevation of the inhibitory curve at negative correlation indicates an effect of the type of regulation.*



random selection of inhibitory (dark gray) and activating (light gray) interactions. No consistent positive correlation of expression levels at the starting point and end point of an activating link or, respectively, an anti-correlation at the starting point and end point of an inhibitory link is observable.

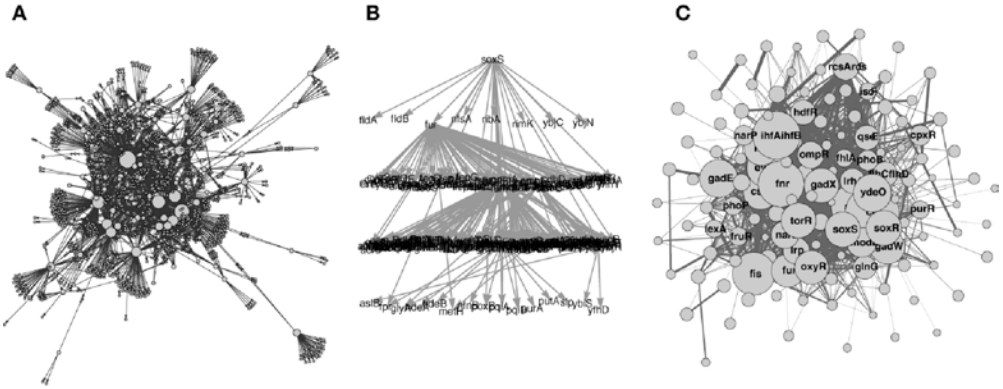
The failure of this naive approach is in agreement with previous findings (Gutierrez et al., 2003; Herrgard et al., 2003) about the low fraction of consistent elements in pairwise regulatory interactions. These studies showed that the consistency between the expression profiles and the TRN is surprisingly low (around 10% for pairwise interactions in Herrgard et al. (2003) and about 40% for complex interactions after a discretization process and sophisticated Boolean rule table application in Gutierrez et al. (2003)). The distributions of correlation values for both activating and inhibiting interactions are very similar (see Figure 4B) with only a slightly elevated amount of large negative correlation for inhibitory interactions. One obvious reason for the absence of clear correlations in pairwise regulatory interactions is the ubiquity of complex promoters, where TFs act cooperatively to reach a transcriptional goal. Another reason might be the vast amount of data where neither the TF, nor the target reach a significant or functional relevant concentration level. Hence, pairwise interactions clearly cannot be separated from the effects of the rest of the network.

The TRN of *E. coli* is mainly acyclic. Hence, it can be decomposed into subnets with one root node and branch nodes, lying topologically downstream in the TRN. Such decomposition into causally connected nodes is one way to address the link between the TRN and expression data beyond pairs of nodes. The largest connected component of the *E. coli* TRN, reconstructed from the latest RegulonDB version 5.6, contains 1257 nodes and 2666 interactions (see Figure 5A), where self-loops have been discarded. This network contains 129 regulators, i.e. nodes with at least one outgoing link to other nodes. Each one of these regulators can be identified as the root node of the corresponding subnet. The *soxS* gene (Wu & Weiss, 1991) for example, responsible for redox stress response (Nunoshiba et al., 1993) has 24 direct target nodes and is regulated by only one other gene, *soxR*. The *soxS* subnet comprises 226 branch nodes and 259 regulatory interactions, organized in five hierarchical layers, as shown in Figure 5B. The hierarchical level of each node is determined by a top down approach: The distance to the root node is calculated with a breadth first algorithm, with the additional constraint that links from lower to higher levels are forbidden. The whole TRN can be decomposed into a subnet graph, with nodes representing subnets and links between subnets indicate an overlapping set of branch nodes. This subnet graph consists of 129 nodes and 1144 links and is shown in Figure 5C.

Practical Example #2: Determining the Relative Contribution of the TRN in the Gene Regulatory Network in the Bacterium *E. Coli*

We can use the conceptual concepts of subnets described above, and their implementation in the RegulonDB database to determine the relative contribution of the TRN compared to all the other control from RNA, proteins, and the metabolic network in gene expression that taken together define the gene regulatory network. We do this in the following way. For any experimental gene deletion, we suppose that the branch nodes in the respective mutant subnet preferentially sense the mutational deletion by transcriptional regulatory connections. By comparing gene expression levels between the wild type and single-gene mutations, we can test the hypothesis that changes occur mostly topologically downstream of the mutated node rather than globally distributed throughout the network. This will then give us an estimate of the role played by the TRN in the total gene regulatory network.

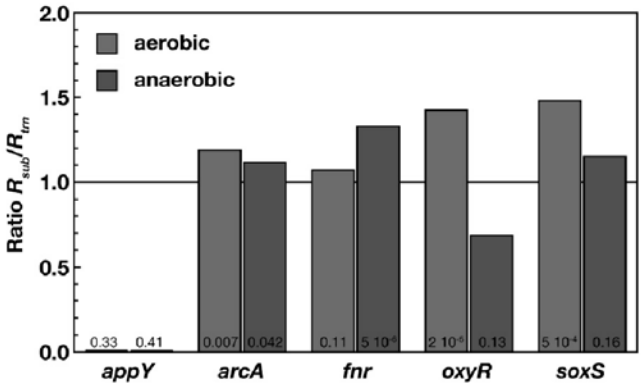
Figure 5. Decomposition of the E. coli TRN into a subnet graph. **A** *The largest connected component of the E. coli TRN comprises 1257 nodes, representing genes, and 2666 interactions, representing TF binding action. The size of the nodes scales with the number of regulated genes.* **B** *The soxS subnet comprises 227 nodes and 259 regulatory interactions, organized into 5 hierarchical levels.* **C** *A full decomposition of the TRN results in a subnet graph with nodes representing subnets and links representing overlapping sets of branch nodes. The size of the nodes scales with the number of genes contained in the respective subnet, the width of the links scales with the relative overlap between the sets of branch nodes.*



The microarray data used in Covert et al. (2004) contains absolute expression profiles for *E. coli* knockout strains of key transcriptional regulators in oxygen response ($\Delta arcA, \Delta appY, \Delta fnr, \Delta oxyR, \Delta soxS$) under aerobic and anaerobic conditions. We compare the expression levels in wild type and mutant strains of all genes under either aerobic or anaerobic conditions. For each gene, each strain, and each condition, at least a triplet of replicated absolute expression values exists. We label a gene as affected, if the p-value of a two-sided t-test of the two distributions of expression values is below 0.05. (Here we tested the effect of each gene individually on its downstream components. An alternative statistical approach would have to be use an FDR multi-stage analysis such as that described by Tuglus and van der Laan, 2008.) We count the relative number of affected genes within the subnet, R_{sub}, and compare it to the relative number of affected genes in the whole network R_{trn}. The ratio of these two observables R_{sub}/R_{trn} measures the causal impact of the mutation within the subnet. The p-value of this ratio, calculated with a null model subnet consisting of randomly selected genes in the TRN without necessarily regulatory interactions, tells about the significance of the measured impact. As a concrete example, let us consider the *soxS* subnet, shown in Figure 5B. Under aerobic growth, 50 out of the 226 subnet genes are affected by the mutation of the root node, *soxS*, resulting in $R_{sub} = 50/226$. In the largest connected component of the TRN, comprising 1257 nodes, 188 are affected by the mutation of *soxS*, $R_{trn} = 188/1257$. Thus, the ratio of these two observables, measuring the relative impact of the *soxS* knockout on the *soxS* subnet, is 1.5. We can estimate the significance of this value by calculating the p-value of this ratio, i.e., the probability that this ratio or an even more extreme value occurs by chance. The p-value of each ratio can be calculated with the hypergeometric distribution. In terms of the example above, it gives the probability that we find 50 affected genes if we randomly select a set of 226 genes from a total number of 1257 genes, containing 188 affected genes.

Figure 6 shows the impact ratios R_{sub}/R_{trn} for all five mutant strains under aerobic and anaerobic conditions. The *appY* subnet comprises only 9 nodes, none of which is affected in either condition.

Figure 6. Impact ratio R_{sub}/R_{trn} for wild type versus mutant strain comparison under aerobic and anaerobic conditions. R_{sub} quantifies the relative number of affected genes within the subnet of the mutant gene, R_{trn} quantifies the relative number of affected genes within the largest connected component of the E. coli TRN. The numbers inside the bars denote the p-value of the calculated ratio. The size of the respective subnets is 9 (appY), 506 (arcA and fnr), 201 (oxyR), and 226 (soxS).



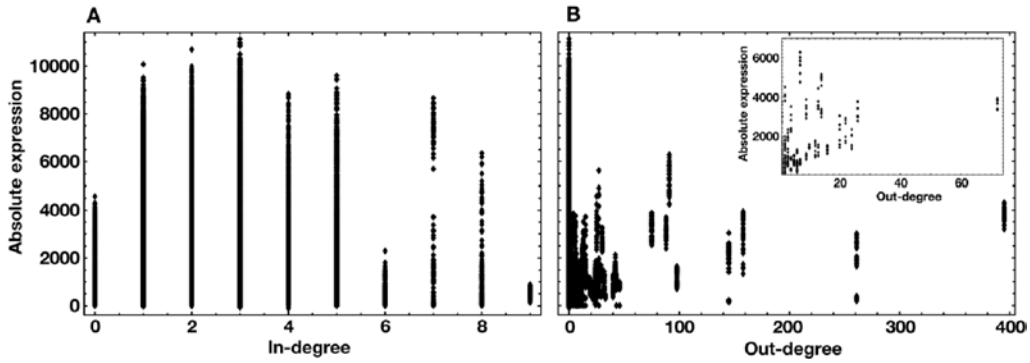
The *arcA* and *fnr* subnets are affected in both conditions, with moderate ratios between 1.1 and 1.3. The *oxyR* subnet is affected under aerobic conditions, which is sensible, since the corresponding gene product OxyR is involved in the response to oxidative stress only (Lynch and Lin, 1996). Interestingly, the *oxyR* subnet is under-affected under anaerobic conditions with a ratio of 0.7, i.e., considerably less genes are affected within the subnet as compared to the whole TRN. This is also in accordance with the role of OxyR: Under anaerobic conditions, the subnet is neither used in the wild type, nor in the mutant strain. Finally, the *soxS* subnet is again affected under aerobic and anaerobic conditions. Although we find sensible values for the subnet affection for four of the investigated five mutant subnets, only half of these values are significant with p-values < 0.01. In all other cases, a large fraction of genes outside the subnet is equally affected: Under anaerobic growth, 29 out of 226 genes within the *soxS* subnet are affected while 111 out 1031 genes in the remaining network equally respond to a *soxS* deletion.

The result of this analysis is that approximately equal number of genes in the subnet and outside the subnet are affected as the expression level of any one gene changes. Therefore, a view where the effect of a mutation solely cascades down the sub-network of transcriptional interactions is obviously inadequate. Additional mechanisms within the TRN must also play a role, such as cooperative effects, like coordinated binding (see, e.g., Hermsen et al. (2006) for a recent computational approach) or DNA supercoiling (Travers and Muskhelishvili, 2005). It also suggests that additional mechanisms beyond the TRN also play an important role.

In summary, this analysis yields the important new finding that the TRN represents only approximately half of the regulatory control in the gene regulatory network. Thus, beside the regulatory role of the transcription factors of the TRN, the other factors that must play an equally important role in gene regulation include RNA, proteins, and the metabolic network.

Practical Example #3: Determining the Functional Dependence of the Expression on the Degree Distribution of the TRN in the bacterium E. Coli

*Figure 7. Gene degree in TRN and expression patterns in E. coli. The absolute expression of the 1262 genes in the E. Coli TRN versus their in-degree (***A***) and their out-degree (***B***). We mapped all 41 aerobic shift experiments onto the inter-regulated genes. Correlations between expression and the number of regulated genes (out-degree) are observable, e.g. genes with many regulated targets genes are constrained to moderate expression. The inset in B shows the mapping of all wild-type expression data on another TRN representation, namely the network of transcriptional regulatory interactions between operons and genes in the E. coli genome as introduced in Shen-Orr et al. (2002). Note that we only display regulatory operons and genes in the inset, i.e. those with out-degree >0.*



We can use the network concepts described above, namely the in-degree and out-degree distributions, to characterize the structure of the TRN. We can then correlate that information with the gene expression data to relate the structure of these networks to their dynamics. We do this by mapping the expression values of the whole data set onto the TRN in *E. Coli*.

While the correlations between expression and in-degree remain rather vague (see Figure 7A), a number of prominent features can be attributed to the correlation of expression and out-degree (Figure 7B): (i) The expression of genes with zero out-degree, that is genes with no regulatory function, covers a very wide range of expression, while the expression of genes with out-degree > 0 is confined to expression levels < 6000. (ii) For genes with 0 < out-degree < 100, two groups clusters seem to emerge, one with moderate expression levels, one with low expression levels. The two groups of nodes appear even clearer in the inset of Figure 7B, where wild type expression data has been mapped on the operon-based TRN of *E. coli*. In this network, which has been introduced in Shen-Orr et al. (2002), nodes represent operons and links represent mutual regulatory interaction between members of two operons. Notably, no discernible pattern of group composition is observable when mapped to gene ontology classes. (iii) For the group of low expressed genes with out-degree > 0, a linear increase of the gene expression levels with the out-degree is observable. (iv) The absolute expression changes only weakly with environmental and genetic variations. Therefore, the out-degree of a gene in the TRN seems to be more influential than the in-degree on the gene's absolute expression.

Summarizing, we find no immediate consistency of expression profiles with activating and inhibitory pairwise interactions in *E. coli*. This is in agreement with the recent observation that a more complicated rule table and an internal consistency of operons are required for detecting an agreement between such data sets (Gutierrez et al., 2003; Herrgard et al., 2003). For single-gene deletions, we find affected nodes both in the subnetwork downstream of the mutation and outside the subnet. While the ratio of affected nodes is generally higher within the sub-network, the large number of affected nodes in the remaining

network suggests that the system deals with perturbations in a cooperative manner beyond solely TRN interactions. Other mapping approaches, like Balázsi et al. (2006), Luscombe et al. (2004), or Marr et al. (2008) supply further functional insights and support this interpretation. Finally, we observe a systematic dependence of gene expression on a topological quantity, namely the out-degree of a node in the TRN, indicating that this is a genuine global network property. We identify classes of genes with specific expression profiles in *E. coli*: non-regulatory genes cover a wide range of absolute expression, while genes with a medium number of regulated target genes separate into two groups of different expression. The article of Grondin et al. (2007) shows similar results and offers a simple explanatory model.

We learn from the analysis of this example that changes in expression levels are widely distributed in the network, even in response to a local perturbation. This shows that gene regulation goes beyond the purely feed-forward structure implied by the TRN. The regulatory information may be fed back into the network by a variety of signals such as physical changes in the DNA conformation or protein-protein interactions resulting in a re-distribution of regulatory signals. Evolutionary adaptation at the TRN network level that leads to an optimal functioning of the organism may explain some features of the distribution pattern of gene expression levels. An interesting example of this perspective is the recent work by Carmi et al. (2006). The observed expression profiles depend on both TF mediated and non-TF mediated regulatory mechanisms, like small non-coding RNAs (Shimoni et al., 2007) or DNA supercoiling (Marr et al., 2008). It seems worthwhile to segregate gene regulations into topological, global and local effects, which can partly be established by the evaluating expression data on TRNs. Indeed, mapping strategies as presented in this chapter may help to quantitatively estimate those non-TF mediated influences on gene expression.

CONCLUSION

Most approaches to understanding the structure and function of the network of how genes regulate the expression of other genes have been to correlate the expression of individual genes and how they vary up and down together under different experimental conditions. That is, to determine the network from the bottom up. Here, we have shown that some important features of that network depend on its global topological properties which can be directly determined from the experimental data. Thus, we can determine important features of the network from the top down. We showed how the statistics of the levels of mRNA reflects the global connectivity pattern of this network. This tells us that the gene regulatory network has a self-similar connection topology with a scaling exponent between 1.5 and 2. By bringing together information about the structure of the transcription regulatory network from the RegulonDB database and its function as measured by the mRNA expression levels recorded by microarrays, we showed that the patterns of mRNA expression depend on the topology of the network, namely on how many other genes any one gene regulates, and on how many genes regulate it. Many different mechanisms are important in the regulation of gene expression including: the physical structure of DNA, the transcription regulatory network defined by the action of transcription factor proteins, RNA interference mechanisms, and protein-protein reaction networks whose proteins can bind back onto DNA and regulate gene expression. By comparing the number of genes that change their expression in the transcription factor subnetwork of a gene after a mutation or a change in an environmental condition, to the number of genes that change their expression outside of this subnetwork, we determined that approximately 1/2 of all the regulatory control of genes is through transcription factor proteins. The remainder of the genetic

regulatory control must therefore be the result of physical changes in DNA structure, RNA editing, or protein-protein interactions. This allows us to establish the fact that the transcription regulatory network is an important component but not the complete story in the control of gene expression.

FUTURE RESEARCH DIRECTIONS

Our current research generates three important challenges for the future: 1) How can we match the local analysis of the small networks of a few interacting genes (such as that presented by Alon, 2007) into the global properties of the networks of large numbers of interacting genes (such as the analysis of Shehadeh et al. 2006, and Liebovitch et al. 2006)? This could be accomplished by both bottom-up approaches trying to synthesize the global dynamics from sets of coupled motifs or by top-down approaches that decompose global dynamics into constituent simpler dynamical systems. 2) The networks of DNA, RNA, metabolism, and proteins all interact with each other. Can we understand the operation of each of these three networks separately, or do we have to treat parts of them, or all of them, together in order to actually understand each one? Over the last several hundred years the trend in biology has been to dissect living things into separate non-interacting pieces and then to try to understand how each piece works. Can we understand how these pieces work separately from each other? The central issue is to determine what features and functions we can meaningfully learn from each of these three networks separately and what features and functions we can understand only from understanding how all these three networks interact together. This issue needs to be approached both experimentally (studying the different behaviors of isolated and whole systems) and theoretically (studying how the global dynamics of a system is different or the same from the dynamics of its constituent systems). 3) A hundred years ago Paul Ehrlich searched for a chemical magic bullet that would cure syphilis and have no other effects (Davis et al., p. 112). But there are no magic bullets, each chemical has both desired effects and undesired “side-effects”. These side effects are the necessary result of the cascade of interactions that the chemical induces in the complex biological regulatory networks. Rather than a fruitless search for magic bullets, perhaps we can use the complexity of these interacting networks to our advantage. If we understand enough about these networks, we can design therapies of multiple chemicals, which will interact within these networks, in just the right way, so as to increase their desired effects and reduce their undesired side effects (Liebovitch et al., 2007). How much do we need to understand about these interacting DNA, RNA, and protein networks in order to predict the effects of multiple inputs to make such a combinatorial multi-component therapy (CMCT) a reality? In our preliminary studies of this issue we have now demonstrated that data from drugs presented one-at-time and pairs-at-time can be used, with an artificial neural network, to accurately compute all the drug interactions for drugs presented 15-at-a-time for even a highly nonlinear network of interactions (Liebovitch et al., 2007). This is an important starting point for developing models of interaction that may lead to such combinatorial multi-component therapy therapies.

REFERENCES

Adamic, L. A., & Huberman, B. A. (1999). Growth dynamics of the World Wide Web. *Nature*, *401*, 131. doi:10.1038/43604

Albert, R., & Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*, 47–97. doi:10.1103/RevModPhys.74.47

Alon, U. (2007). Network motifs: Theory and experimental approaches. *Nature Genetics, 8*, 450–461. doi:10.1038/nrg2102

Amaral, L. A. N., Diaz-Guilera, A., Moreira, A. A., Goldberger, A. L., & Lipsitz, L. A. (2004). Emergence of complex dynamics in a simple model of signaling networks. *Proceedings of the National Academy of Sciences of the United States of America, 101*, 15551. doi:10.1073/pnas.0404843101

Arenas, A., Diaz-Guilera, A., & Perez-Vicente, C. J. (2006). Synchronization reveals topological scales in complex networks. *Physical Review Letters, 96*, 114102. doi:10.1103/PhysRevLett.96.114102

Arnone, M. I., & Davidson, E. H. (1997). The hardwiring of development: Organization and function of genomic regulatory systems. *Development, 124*, 1851–1864.

Atay, F. M., & Biyikoglu, T. (2005). Graph operations and synchronization of complex networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 72*, 016217. doi:10.1103/PhysRevE.72.016217

Balázsi, G., Barabási, A. L., & Oltvai, Z. N. (2005). Topological units of environmental signal processing in the transcriptional regulatory network of Escherichia coli. *Proceedings of the National Academy of Sciences of the United States of America, 102*, 7841. doi:10.1073/pnas.0500365102

Banerjee, A., & Jost, J. (2007). Spectral plots and the representation and interpretation of biological data. *Theory in Biosciences, 126*, 15–21. doi:10.1007/s12064-007-0005-9

Barabasi, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286*, 509–512. doi:10.1126/science.286.5439.509

Basso, K., Margolin, A., Stolovitzky, G., Klein, U., Dalla-Favera, R., & Califano, A. (2005). Reverse engineering of regulatory networks in human B cells. *Nature Genetics, 37*, 382–390. doi:10.1038/ng1532

Bornholdt, S. (2005). Less is more in modeling large genetic networks. *Science, 310*, 449. doi:10.1126/science.1119959

Bower, J. M., & Bolouri, H. (Eds.). (2001). *Computational modeling of genetic and biochemical networks.* Cambridge, MA: MIT Press.

Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene networks: How to put the function in genomics. *Trends in Biotechnology, 20*, 467–472. doi:10.1016/S0167-7799(02)02053-X

Carmi, S., Levanon, E. Y., Havlin, S., & Eisenberg, E. (2006). Connectivity and expression in protein networks: Proteins in a complex are uniformly expressed. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 73*, 031909. doi:10.1103/PhysRevE.73.031909

Covert, M. W., Knight, E. M., Reed, J. L., Herrgard, M. J., & Palsson, B. O. (2004). Integrating high-throughput and computational data elucidates bacterial networks. *Nature, 429*, 92–96. doi:10.1038/nature02456

Crick, F. H. C. (1958). In *Symp. Soc. Exp. Biol., The Biological Replication of Macromolecules, XII*, 138.

Davidich, M., & Bornholdt, S. (2007). (in press). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE*.

Davis, D. D., Dulbecco, R., Eisen, H. N., & Ginsberg, H. S. (Eds.). (1980). *Microbiology*. Philadelphia, PA: Harper & Row.

Dobrin, R., Beg, Q. K., Barabási, A.-L., & Oltvai, Z. N. (2004). Aggregation of topological motifs in the Escherichia coli transcriptional regulatory network. *BMC Bioinformatics, 5*, 10. doi:10.1186/1471-2105-5-10

Erdös, P., & Rényi, A. (1959). On random graphs. *Publicationes Mathematicae (Debrecen), 6*, 290–297.

Fasken, M. B., & Corbett, A. H. (2005). Process or perish: Quality control in mRNA biogenesis. *Nature Structural & Molecular Biology*, *12*, 482–488. doi:10.1038/nsmb945

Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *301*(5629), 102–105. doi:10.1126/science.1081900

Glasner, J. D., Liss, P., Plunkett, G. III, Darling, A., Prasad, T., & Rusch, M. (2003). ASAP, a systematic annotation package for community analysis of genomes. *Nucleic Acids Research*, *31*(1), 147–151. doi:10.1093/nar/gkg125

Graham, I., & Matthai, C. C. (2003). Investigation of the forest-fire model on a small-world network. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *68*, 036109. doi:10.1103/PhysRevE.68.036109

Grondin, Y., Raine, D. J., & Norris, V. (2007). The correlation between architecture and mRNA abundance in the genetic regulatory network of escherichia coli. *BMC Systems Biology*, *1*, 30. doi:10.1186/1752-0509-1-30

Guelzim, N., Bottani, S., Bourgine, P., & Kepes, F. (2002). Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics*, *31*(1), 60–63. doi:10.1038/ng873

Gutierrez-Rios, R. M., Rosenblueth, D. A., Loza, J. A., Huerta, A. M., Glasner, J. D., Blattner, F. R., & Collado-Vides, J. (2003). Regulatory network of Escherichia coli: Consistency between literature knowledge and microarray profiles. *Genome Research*, *13*(11), 2435–2443. doi:10.1101/gr.1387003

Hermsen, R., Tans, S., & ten Wolde, P. R. (2006). Transcriptional regulation by competing transcription factor modules. *PLoS Computational Biology*, *2*, e164. doi:10.1371/journal.pcbi.0020164

Herrgard, M. J., Covert, M. W., & Palsson, B. O. (2003). Reconciling gene expression data with known genome-scale regulatory network structures. *Genome Research*, *13*(11), 2423–2434. doi:10.1101/gr.1330003

Huberman, B. A., Pirolli, P. L. T., Pitkow, J. E., & Luose, R. M. (1998). Strong regularities in World Wide Web surfing. *Science*, *280*, 95–97. doi:10.1126/science.280.5360.95

Ideker, T., Thorsson, V., Ranish, J., Christmas, R., Buhler, J., & Eng, J. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, *292*(5518), 929. doi:10.1126/science.292.5518.929

Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., & Barkai, N. (2002). Revealing modular organization in the yeast transcriptional network. *Nature Genetics*, *31*, 370–377.

Jacob, F., & Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, *3*, 318–356.

Jaenisch, R., & Bird, A. (2003). Epigenetic regulation of gene expression: How the genome integrates intrinsic and environmental signals. *Nature Genetics*, *33*(Suppl), 245–254. doi:10.1038/ng1089

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*, 437. doi:10.1016/0022-5193(69)90015-0

Khodursky, A. B., & Bernstein, J. A. (2003). Life after transcription--revisiting the fate of messenger RNA. *Trends in Genetics*, *19*, 113–115. doi:10.1016/S0168-9525(02)00047-1

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, *46*, 604–632. doi:10.1145/324133.324140

Klemm, K., & Bornholdt, S., S. (2005). Topology of biological networks and reliability of information processing. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 18414. doi:10.1073/pnas.0509132102

Lee, D.-S., & Rieger, H. (2007). Comparative study of the transcriptional regulatory networks of E. coli and yeast: Structural characteristics leading to marginal dynamic stability. *Journal of Theoretical Biology*, *248*(4), 618–626. doi:10.1016/j.jtbi.2007.07.001

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., & Gerber, G. K. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, *298*, 799. doi:10.1126/science.1075090

Levine, M., & Tjian, R. (2003). Transcription regulation and animal diversity. *Nature*, *424*, 147–151. doi:10.1038/nature01763

Liebovitch, L. S., Jirsa, V. K., & Shehadeh, L. A. (2006). Structure of genetic regulatory networks: Evidence for scale free networks. In M. M. Nowak (Ed.), *Complexus mundi: Emergent patterns in nature* (pp. 1-8). Singapore: World Scientific.

Liebovitch, L. S., Tsinoremas, N., & Pandya, A. (2007). Developing combinatorial multicomponent therapies (CMCT) of drugs that are more specific and have fewer side effects than traditional one drug therapies. *Nonlinear Biomedical Physics*, *1*, 11. doi:. doi:10.1186/1753-4631-1-11

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A., & Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, *431*, 308–312. doi:10.1038/nature02782

Lynch, A. S., & Lin, E. C. C. (1996). Responses to molecular oxygen. In *Escherichia coli and salmonella: Cellular and molecular biology* (pp. 1526–1538). Washington, D. C.: American Society for Microbiology, 2nd edition.

Ma, H.-W., Buer, J., & Zeng, A.-P. (2004). Hierarchical structure and modules in the escherichia coli transcriptional regulatory network revealed by a new top-down approach. *BMC Bioinformatics, 5,* 199. doi:10.1186/1471-2105-5-199

Marr, C., Geertz, M., Hütt, M.-T., & Mushkhelishvili. (2008). Dissecting the logical types of network control in gene expression profiles. *BMC Sys. Biol.*, 2, 18.

Marr, C., & Hütt, M.-Th. (2005). Topology regulates pattern formation capacity of binary cellular automata on graphs. *Physica A, 354,* 641–662. doi:10.1016/j.physa.2005.02.019

Marr, C., & Hütt, M.-Th. (2006). Similar impact of topological and dynamic noise on complex patterns. *Physics Letters. [Part A], 349,* 302–305. doi:10.1016/j.physleta.2005.08.096

Mata, J., Marguerat, S., & Bahler, J. (2005). Post-transcriptional control of gene expression: A genome-wide perspective. *Trends in Biochemical Sciences, 30,* 506–514. doi:10.1016/j.tibs.2005.07.005

McAdams, H., & Arkin, A. (1997). Stochastic mechanism in gene expression. *Proceedings of the National Academy of Sciences of the United States of America, 94,* 814. doi:10.1073/pnas.94.3.814

Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., & Ayzenshtat, I. (2004). Superfamilies of evolved and designed networks. *Science, 303,* 1538. doi:10.1126/science.1089167

Moreira, A. A., Mathur, A., Diermeier, D., & Amaral, L. A. N. (2004). Efficient system-wide coordination in noisy environments. *Proceedings of the National Academy of Sciences of the United States of America, 101,* 12085. doi:10.1073/pnas.0400672101

Moreno, Y., Pastor-Satorras, R., & Vespignani, A. (2002). Epidemic outbreaks in complex heterogeneous networks. *The European Physical Journal B, 26,* 521.

Müller-Linow, M., Hilgetag, C., & Hütt, M.-Th. (2008). (in press). Organization of excitable dynamics in hierarchical biological networks. *PLoS Computational Biology*.

Müller-Linow, M., Marr, C., & Hütt, M.-Th. (2006). Topology regulates synchronization patterns in excitable dynamics on graphs. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 74,* 016112. doi:10.1103/PhysRevE.74.016112

Nishikawa, T., & Motter, A. E. (2006). Synchronization is optimal in nondiagonalizable networks . *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 73,* 065106. doi:10.1103/PhysRevE.73.065106

Nunoshiba, T., Hidalgo, E., Li, Z., & Demple, B. (1993). Negative autoregulation by the escherichia coli soxs protein: A dampening mechanism for the soxrs redox stress response. *Journal of Bacteriology, 175*(22), 7492–7494.

Parker, R., & Song, H. (2004). The enzymes and control of eukaryotic mRNA turnover. *Nature Structural & Molecular Biology, 11,* 121–127. doi:10.1038/nsmb724

Roth, S. Y., Denu, J. M., & Allis, C. D. (2001). Histone acetyltransferases. *Annual Review of Biochemistry, 70*, 81–120. doi:10.1146/annurev.biochem.70.1.81

Roxin, A., Riecke, H., & Solla, S. A. (2004). Self-sustained activity in a small-world network of excitable neurons. *Physical Review Letters, 92*, 198101. doi:10.1103/PhysRevLett.92.198101

Salgado, H., Gama-Castro, S., Peralta-Gil, M., Diaz-Peredo, E., Sanchez-Solano, F., & Santos-Zavaleta, A. (2006). RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research, 34*, D394–D397. doi:10.1093/nar/gkj156

Shehadeh, L., Liebovitch, L. S., & Jirsa, V. K. (2006). The structure of genetic networks determined from mRNA levels measured by cDNA microarrays . *Physica A, 364*, 297–314. doi:10.1016/j.physa.2005.08.069

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics, 31*, 64–68. doi:10.1038/ng881

Shimoni, Y., Friedlander, G., Hetzroni, G., Niv, G., Altuvia, S., Biham, O., & Margalit, H. (2007). Regulation of gene expression by small non-coding RNAs: A quantitative view. *Molecular Systems Biology*, 3.

Sporns, O., & Tononi, G. (2007). Structural determinants of functional brain connectivity. In V. K. Jirsa & A. R. M. McIntosh (Eds.), *Handbook of brain connectivity.* Springer.

Strogatz, S. H. (2001). Exploring complex networks. [London.]. *Nature, 410*, 268–277. doi:10.1038/35065725

Thanos, D., & Maniatis, T. (1995). Virus induction of human IFN beta gene expression requires the assembly of an enhanceosome. *Cell, 83*, 1091–1100. doi:10.1016/0092-8674(95)90136-1

Thieffry, D., Huerta, A. M., Pèrez-Rueda, E., & Collado-Vides, J. (1998). From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in escherichia coli. *BioEssays, 20*, 433–440. doi:10.1002/(SICI)1521-1878(199805)20:5<433::AID-BIES10>3.0.CO;2-2

Travers, A., & Muskhelishvili, G. (2005). DNA supercoiling-a global transcriptional regulator for enterobacterial growth? *Nature Reviews Microbiology, 3*, 157–169. doi:10.1038/nrmicro1088

Tsang, J., Zhu, J., & van Oudenaarden, A. (2007, June). MicroRNA-mediated feedback and feedforward loops are recurrent network motifs in mammals. *Molecular Cell, 26*(5), 753–767. doi:10.1016/j.molcel.2007.05.018

Tuglus, C., & van der Laan, M. (2008). FDR controlling procedure for mult-stage analysis. *U. C. Berkeley Div. Biostsat.* Working paper series, paper 239.

von Neumann, J. (2001). In A. H. Taub (Ed.), *J. von Neumann, Collected Works*, 5, 288. New York: Macmillan.

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature, 393*, 440–442. doi:10.1038/30918

Wigner, E. (1958). On the distribution of the roots of certain symmetric matrices. *The Annals of Mathematics*, *67*, 325–328. doi:10.2307/1970008

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics*, *55*, 601. doi:10.1103/RevModPhys.55.601

Wray, G. A., Hahn, M. W., Abouheif, E., Balhoff, J. P., Pizer, M., Rockman, V., & Romano, L. A. (2003). The evolution of transcriptional regulation in eukaryotes. *Molecular Biology and Evolution*, *20*, 1377–1419. doi:10.1093/molbev/msg140

Wu, J., & Weiss, B. (1991). Two divergently transcribed genes, soxr and soxs, control a superoxide response regulon of escherichia coli. *Journal of Bacteriology*, *173*(9), 2864–2871.

Yook, S.-H., & Meyer-Ortmanns, H. (2006). Synchronization of Rössler oscillators on scale-free topologies. *Physica A*, *371*, 781–789. doi:10.1016/j.physa.2006.04.116

Yu, H., & Gerstein, M. (2006). Genomic analysis of the hierarchical structure of regulatory networks. *Proceedings of the National Academy of Sciences of the United States of America*, *103*, 14724–14731. doi:10.1073/pnas.0508637103

ADDITIONAL READING

Barabasi, A.-L. (2002). *Linked: The new science of networks.* Cambridge, MA: Perseus Publishing.

Bower, J. M., & Bolouri, H. (Eds.). (2001). *Computational modeling of genetic and biochemical networks.* Cambridge, MA: MIT Press.

Klipp, E., Herwig, R., Kowald, A., Wierling, C., & Leharch, H. (2005). *Systems biology in practice.* Berlin: Wiley-VCH.

Kriete, A., & Eils, R. (Eds.). (2006). *Computational systems biology.* New York: Academic Press.

Newman, M., Barabasi, A.-L., & Watts, D. J. (Eds.). (2006). *The structure and dynamics of networks.* Princeton, NJ: Princeton University Press.

KEY TERMS AND DEFINITIONS

Connectivity Matrix: This matrix, w_{ij}, defines the strengths of the links between nodes i and j in a network.

Degree Distribution: The statistical distribution of the number of nodes that connect to each node in a network.

Gene Regulatory Network: The network of interactions that define how genes regulate each other through DNA, RNA, and protein interactions.

Motif: A network of interactions between genes that occurs more often than those expected of purely random connections.

Network: A set of nodes and links between them. In gene regulatory networks, the nodes are genes and the links between them are the DNA, RNA, and protein interactions between the genes.

Probability Density Function (PDF): The probability that values between x and x + dx are found with probability p(x).

Sub-Net: All of the genes connected by transcription factors downstream from a given gene.

Transcription Factor: A protein expressed by a gene that binds to DNA and regulates the expression of that gene or other genes.

Transcription Regulatory Network (TRN): The network of gene interactions mediated by the transcription factors expressed by genes that regulate other genes.

Chapter 18
Generalized Boolean Networks: How Spatial and Temporal Choices Influence Their Dynamics

Christian Darabos
University of Lausanne, Switzerland; University of Turin, Italy

Mario Giacobini
University of Torino, Italy

Marco Tomassini
University of Lausanne, Switzerland

ABSTRACT

Random Boolean Networks (RBN) have been introduced by Kauffman more than thirty years ago as a highly simplified model of genetic regulatory networks. This extremely simple and abstract model has been studied in detail and has been shown capable of extremely interesting dynamical behavior. First of all, as some parameters are varied such as the network's connectivity, or the probability of expressing a gene, the RBN can go through a phase transition, going from an ordered regime to a chaotic one. Kauffman's suggestion is that cell types correspond to attractors in the RBN phase space, and only those attractors that are short and stable under perturbations will be of biological interest. Thus, according to Kauffman, RBN lying at the edge between the ordered phase and the chaotic phase can be seen as abstract models of genetic regulatory networks. The original view of Kauffman, namely that these models may be useful for understanding real-life cell regulatory networks, is still valid, provided that the model is updated to take into account present knowledge about the topology of real gene regulatory networks, and the timing of events, without loosing its attractive simplicity. According to present data, many biological networks, including genetic regulatory networks, seem, in fact, to be of the scale-free type. From the point of view of the timing of events, standard RBN update their state synchronously. This assumption is open to discussion when dealing with biologically plausible networks. In particular, for genetic regulatory networks, this is certainly not the case: genes seem to be expressed in different parts of the network at different times, according to a strict sequence, which depends on the particular

DOI: 10.4018/978-1-60566-685-3.ch018

network under study. The expression of a gene depends on several transcription factors, the synthesis of which appear to be neither fully synchronous nor instantaneous. Therefore, we have recently proposed a new, more biologically plausible model. It assumes a scale-free topology of the networks and we define a suitable semi-synchronous dynamics that better captures the presence of an activation sequence of genes linked to the topological properties of the network. By simulating statistical ensembles of networks, we discuss the attractors of the dynamics, showing that they are compatible with theoretical biological network models. Moreover, the model demonstrates interesting scaling abilities as the size of the networks is increased.

INTRODUCTION TO RANDOM BOOLEAN NETWORKS

Gene regulatory networks are formed by genes, messenger RNA, and proteins. The interactions between these elements include transcription, translation, and transcriptional regulation (Albert, 2001). The processes are extremely complex and we are just beginning understanding them in detail. However, it is possible, and useful, to abstract many details of the particular kinetic equations in the cell and focus on the system-level properties of the whole network dynamics. This Complex Systems Biology approach, although not strictly applicable to any given particular case, may still provide interesting general insight.

Random Boolean Networks (RBNs) have been introduced by Kauffman more than thirty years ago (Kauffman, 1969) as a highly simplified model of genetic regulatory networks (GRNs). RBNs have been studied in detail by analysis and by computer simulations of statistical ensembles of networks and it has been shown to be capable of surprising dynamical behavior.

In the last decade, a host of new findings and the increased availability of biological data has changed our understanding of the structure and functioning of GRNs. In spite of this, we believe that the original view of Kauffman is still valid, provided that the model is updated to take into account the new knowledge about the topological structure and the timing of events of real gene regulatory networks without loosing its attractive simplicity. Following these guidelines, our aim in this work is to describe and test a new model that we call Generalized Boolean Networks (GBNs), which includes, at a high level of abstraction, structures and mechanisms that are hopefully closer to the observed data.

Adhering to the original Kauffman's view that attractors of the dynamics of RBNs are the important feature and that they roughly correspond to cell types, we will discuss the results of the systems ability to relax into stable cycles.

In Kauffman's RBNs (known as Classical RBNs) with N nodes, a node represents a gene and is modeled as an on-off device, meaning that a gene is expressed if it is on (1), and it is not otherwise (0). Each gene receives K randomly chosen inputs from other genes. From a simplistic point of view, the combined effect of proteins produced by genes g_1 to g_K attaching to a *mRNA* binding site, thus either promoting or repressing the activity of gene g, can be seen as a direct effect of a function $f(g_1,\ldots,g_K,g,t) \rightarrow g^{t+1}$. In this case, we allow g to be one of the arguments of the gene update function f, thus permitting self-regulation. If we assume all genes are Boolean nodes, we can define the activity of any gene at time $t+1$ as the result of a Boolean function of each of the gene's entries at time t.

Initially, one of the 2^{2^K} possible Boolean functions of K inputs is assigned at random to each gene. The network dynamics is discrete and synchronous: at each time step all nodes simultaneously examine

their inputs, evaluate their Boolean functions, and find themselves in their new states at the next time step.

More precisely, the *local transition rule* φ, which is one the $2^{2^{K+1}}$ possible Boolean functions of K inputs form the neighboring nodes plus that of the node itself, thus possibly implementing a biological situation where a gene regulates itself:

$$\varphi : \Sigma^{K+1} \rightarrow \Sigma,$$

maps the state $s_i \in \Sigma = \{0,1\}$ of a given node i into another state from the set Σ, as a function of the states of the nodes that send inputs to i.

For a finite-size system of size N (such as those treated herein) a *configuration* $C(t)$ of the RBN at time t is defined by the binary string:

$$C(t) = (s_0(t), s_1(t), \ldots, s_{N-1}(t)),$$

where $s_i(t) \in \Sigma$ is the state of node i at time t. The progression of the RBN in time is then given by the iteration of the *global mapping*, also called *evolution operator* Φ:

$$\Phi : C(t) \rightarrow C(t+1), ++ t = 0,1,\ldots$$

through the simultaneous application at each node of the local transition rule φ. The global dynamics of the RBN can be described as a directed graph, referred to as the CA's *phase space*. Over time, the system travels through its phase space, until a point or cyclic attractor is reached whence either it will remain in that point attractor forever, or it will cycle through the states of the periodic attractor. Since the system is finite and deterministic, this will happen at most after 2^N time steps.

This extremely simple and abstract model has been studied in detail by analysis and by computer simulations of statistical ensembles of networks and it has been shown to be capable of surprising dynamical behavior. Complete descriptions can be found in (Kauffman, 1993; Aldana, Coppersmith & Kadanoff, 2003). We summarize the main results here.

First of all, it has been found that, as some parameters are varied such as K, or the probability p of expressing a gene, i.e. of switching on the corresponding node's state, the RBN can go through a phase transition. Indeed, for every value of p, there is a critical value of connectivity:

$$K_c(p) = [2p(1-p)]^{-1}$$

such that for values of K below this critical value $K_c(p)$ the system is in the ordered regime, while for values of K above this limit the system is said to be in the chaotic regime.

The regimes can be differentiated according to the proportion of nodes that are actively participating in an attractor by flipping their states "often". In other words, assume that we can define two categories for the nodes of a system in an attractor: *frozen* and *twinkling* (Kauffman, 2000). Frozen nodes are those whose state remains unchanged for a long time, say fifty time steps. On the contrary, twinkling ones change their state frequently. In the ordered regime, the proportion of frozen nodes grows linearly with

the network's size N, and a vast majority of the nodes are frozen. In the chaotic regime a majority remain twinkling. Finally at the critical regime, or so-called edge-of-chaos, the number of twinkling and frozen nodes is comparable. Another critical feature distinguishing the ordered form the chaotic regime is that in the first one, the lengths of the state cycle attractors scales palynomially with the size of the network, where as in the chaotic regime, it grows exponentially.

In classical RBNs $K_c(p) = 2$, for $p = 0.5$, corresponds to the edge between the ordered and the chaotic regime, systems where $K < 2$ are in the ordered regime, and $K > 2$ means that the system is in the chaotic phase for $p = 0.5$. Kauffman found that for $K = 2$ the size distribution of perturbations in the networks is a power law with finite cutoff that scales as the square root of N. Thus perturbations remain localized and do not percolate through the system. The mean cycle length scales at most linearly with N for $K = 2$. Kauffman's suggestion was that cell types correspond to attractors in the RBN phase space, and only those attractors that are short (between 2 and a few tens or hundreds of states) and stable under perturbations will be of biological interest. Thus, according to Kauffman, $K = 2$ RBNs lying at the edge between the ordered phase and the chaotic phase can be seen as abstract models of genetic regulatory networks.

For the sake of completeness, let us mention that the "discrete'' approach to the high-level description of genetic regulatory networks is not the only possible one. A more realistic description is obtained through the use of a "continuous-state'' model. In the latter, the levels of messenger RNA and proteins are assumed to be continuous functions of time instead of on/off variables. The system evolution is thus represented by sets of differential equations modeling the continuous variation of the components concentration. Here we focus on the discrete approach, but the interested reader can find more information on the continuous models in (Edwards & Glass, 2006), for instance.

RBNs are interesting in their own as complex dynamical systems and have been throughly studied as such using the concepts and tools of statistical mechanics (Derrida & Pomeau, 1969; Aldana, Coppersmith & Kadanoff, 2003). There is nothing wrong with this; however, we believe that the original view of Kauffman, namely that these models may be useful for understanding real cell regulatory networks, is still a valid one, provided that the model is updated to take into account present knowledge about the topology of real gene regulatory networks, and the timing of events, without loosing its attractive simplicity.

FROM RANDOM TO GENERALIZED BOOLEAN NETWORKS

In this section, we first describe and comment the main assumptions implied in Kauffman's RBNs. Following this, we propose some modifications that, in our opinion, should bring the model closer to known facts about genetic regulatory networks, without loosing the attractive simplicity of classical RBNs.

Kauffman's RBN model rests on three main assumptions:

- The nodes implement Boolean functions and their state is either on or off;
- The nodes that affect a given node in the network are randomly chosen and are a fixed number;
- The dynamics of the network is synchronous in time.

Discrete State Approach

The binary state simplification could seem extreme but actually it represents quite well "threshold phenomena'' in which variables of interest suddenly change their state, such as neurons firing or genes being switched on or off. This can be understood since the sigmoidal functions one finds in the continuous differential equation approach (Edwards & Glass, 2006) actually do reduce to threshold gates in the limit, and it is well known that Boolean functions can be constructed from one or more threshold gates (Hassoun, 1995). So, in the interest of simplicity, our choice is to keep the discrete Boolean model for the states of the nodes and the functions implemented at each node.

Random Networks

RBNs are directed random networks. The edges have an orientation because they represent a chemical influence from one gene to another, and the graphs are random because any node is as likely to be connected to any other node in an independent manner. There are two main types of RBNs, one in which the connections are random but the degree is fixed and a more general one in which only the average connectivity is fixed. Random graphs with fixed connectivity degree were a logical generic choice in the beginning, since the exact couplings in actual genetic regulatory networks were largely unknown. Today it is more open to criticism since it does not correspond to what we know about the topology of biological networks. In fact, many biological networks, including genetic regulatory networks, seem to be of the scale-free type or of a hierarchical type (Vázquez, Dobrin, Sergi, Eckmann, Oltvai & Barabàsi, 2004; Albert, 2005; Christensen, Gupta, Maranas, Albert, 2007) but not random, according to present data, as far as the *output* degree distribution is concerned. The *degree distribution function* $p(k)$ of a graph represents the probability that a randomly chosen node has degree k (Newman, 2003). For directed graphs, there are two distributions, one for the outgoing edges $p_{out}(k)$ and another for the incoming edges $p_{in}(k)$. The *input* degree distributions seem to be close to normal or exponential instead. A scale-free distribution for the degree means that $p(k)$ is a power law $P(k) : k^{-\gamma}$, with γ usually but not always between 2 and 3. In contrast, random graphs have a Poisson degree distribution $p(k); \bar{k}^k e^{-\bar{k}} / k!$, where $\bar{k}$ is the mean degree, or a delta distribution as in a classical fixed-degree RBN. Thus the low fixed connectivity suggested by Kauffman ($K : 2$) for candidate stable systems is not found in such degree-heterogeneous networks, where a wide connectivity range is observed instead. The consequences for the dynamics may be important, since in scale-free graphs there are many nodes with low degree and a low, but not vanishing, number of highly connected nodes (Albert & Barabasi, 2002; Newman, 2003).

The first work that we are aware of using the scale-free topology for Boolean networks dynamics is (Oosawa & Savageaui, 2002). Oosawa and Savageau took *Escherichia coli* as a model for their scale-free nets with an average input degree $\bar{k}$ of two. But, although interesting in this particular case, this is too limited as most other known networks or network fragments have higher connectivity levels. What is needed are models that span the range of observed connectivities.

Along this line, Aldana then presented a detailed analysis of a model Boolean network with scale-free topology (Aldana, 2003). He has been able to define a phase space diagram for scale-free boolean networks, including the phase transition from ordered to chaotic dynamics, as a function of the power law exponent γ. He also made exhaustive simulations for several relatively small values of N, the network size. In our model we have thus adopted networks with a scale-free output distribution, and a

Poissonian input distribution, as this seems to be at least close to the actual topologies. In section 3 we shall give details on the construction of suitable graphs of this type for our simulations.

Time Evolution

Standard RBN update their state synchronously (SU). This assumption simplifies the analysis, but it is open to discussion if the network has to be biologically plausible (Edwards & Glass, 2006). In particular, for genetic regulatory networks, this is certainly not the case, as many recent experimental observations tend to prove. Rather, genes seem to be expressed in different parts of the network at different times, according to a strict sequence (Davidson, 2002). Thus a kind of serial, asynchronous update sequence seems to be needed. Asynchronous dynamics must nevertheless be further qualified, since there are many ways for serially updating the nodes of the network.

Two types of asynchronous updates are commonly used. In the first, a random permutation of the nodes is drawn and the nodes are updated one at a time in that order. At the next update cycle, a fresh permutation is drawn and the cycle is repeated. Let us call this policy *Random Permutation Update* (RPU). In a second often used policy, the next cell to be updated is chosen at random with uniform probability and with replacement. This is a good approximation of a continuous-time Poisson process, and it will be called *Uniform Update* (UU).

Several researchers have investigated the effect of asynchronous updating on classical RBN dynamics in recent years (Harvey & Bossomaier, 1997; Mesot & Teuscher, 2003; Gershenson, 2004). Harvey and Bossomayer studied the effect of random asynchronous updating on some statistical properties of network ensembles, such as cycle length and number of cycles, using both RPU and UU (Harvey & Bossomaier, 1997). They found that many features that arise in synchronous RBN do not exist, or are different in non-deterministic asynchronous RBN. Thus, while point attractors do persist, there are no true cyclic attractors, only so-called loose ones and states can be in more than one basin of attraction. Also, the average number of attractors is very different from the synchronous case: even for $K = 2$ or $K = 3$, which are the values that characterize systems at the edge of chaos, there is no correspondence between the two dynamics.

Mesot and Teuscher studied the critical behavior of asynchronous RBN and concluded that they do not have a critical connectivity value analogous to synchronous RBN and they behave, in general, very differently from the latter, thus confirming in another way the findings of (Harvey & Bossomaier, 1997).

Gershenson (2004) extended the analysis and simulation of asynchronous RBN by introducing additional update policies in which specific groups of nodes are updated deterministically. He found that all types of networks have the same point attractors but other properties, such as the size of the attractor basins and the cyclic attractors do change.

Considering the above reults and what is known experimentally about the timing of events in genetic networks we conclude, along with with Mesot & Teuscher (2003), that neither fully synchronous nor completely random asynchronous network dynamics are suitable models. Synchronous update is implausible because events do not happen all at once, while completely random dynamics does not agree with experimental data on gene activation sequences and the model does not show stable cyclic attractors of the right size. For this reason, in the following section 3 we propose a new quasi-synchronous node update scheme, which we believe is closer to reality.

SEMI-SYNCHRONOUS GENERALIZED BOOLEAN NETWORKS

In this section we first present the methodology for constructing our model networks, and then we describe our new method for updating the nodes' states.

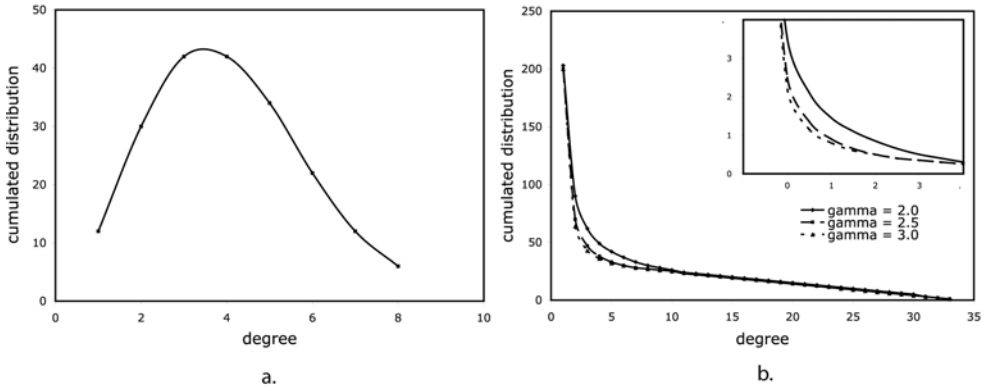
Construction of Model Networks

As said above, Kauffman's RBN are directed graphs. Let's suppose that each node i ($i \in \{1,\ldots,N\}$) receives k_i^{in} inputs and projects a link to other k_i^{out} nodes, i.e. there are k_i^{out} nodes in the graph that receive an input from node i. Among the N nodes of the graph, the distribution $p_{in}(k)$ of the input connections is not necessarily the same of the distribution of the output connections $p_{out}(k)$. In fact, and as anticipated in the preceding section, according to present data many biological networks, including genetic regulatory networks, suggest a scale-free output distribution and a Poissonian or exponential input distribution (Vázquez, Dobrin, Sergi, Eckmann, Oltvai & Barabàsi, 2004; Albert, 2005; Christensen, Gupta, Maranas, Albert, 2007). Whether $p_{in}(k)$ is Poissonian or exponential is almost immaterial for both distributions have a tail that decays quickly, although the Poissonian distribution does so even faster than the exponential, and thus both have a clear scale for the degree. On the other hand, $p_{out}(k)$ is very different, with a fat tail to the right, meaning that there are some nodes in the network that influence many other nodes.

With the intention of following as close as possible the most current hypotesis on the topic of biological networks topologies, we are using networks that follow a scale-free (power-law) output distribution and a Poisson input distribution, see Figure 1. We will compare experimental results of our model with those of Boolean Networks on random toplogies. Our method for generating directed graphs following these degree distributions is a variant of the so-called *configuration model* (Newman, 2003). We call it the *Modified Configuration Model* (MCM) (Tomassini, Giacobini & Darabos, 2007).

First, we must set a few constraints in order to be able to construct connected networks and compare their behavior with that of random topologies:

Figure 1. (a) Poisson input and (b) power-law output degrees cumulated distributions of generated networks of size $N = 200$ and $\gamma = 2.0$, $\gamma = 2.5$ and, $\gamma = 3.0$, obtained using the Modified Configuration Model for $\overline{k} = 4$. Distributions are discrete and finite; the continuous lines are just a guide for the eye.



- the number of vertices N in a network is fixed and constant;
- in the case of a power-law distribution, the exponent γ is predefined in order to have the Boolean Network in a specific regime (order, chaos or edge of chaos);
- the average input degree has to match exactly the average output degree so that each egde has a source and a target vertex. Therefore, we actually fix the average degrees of the distributions. In our case $\overline{k}_{in} = \overline{k}_{out} = 4$;
- the minimum and maximum input and output degree of each node is fixed. In our case $k^{in}_{min} = k^{out}_{min} = 1$ and $k^{in}_{max} = k^{out}_{max} = 50$.

Knowing the desired average degrees $\overline{k}$ and network size N, we can compute the number of edges E in the netwok as $E = \overline{k} \cdot N$. In addition, the output degree distribution must follow the power-law function:

$$f(k) = a \cdot k^{-\gamma} + b$$

and the output distribution must follow the Poisson function

$$g(k) = c \cdot \frac{\overline{k}^{k} \cdot e^{-\overline{k}}}{k!}$$

for all $k \in \{k_{min}, k_{max}\}$. The constants a, b, and c are to be defined so that the distributions satisfy all constraints specified above. In this case, we exhaustively enumerate all possible combinations of a (where $a > 0$ and $a \cdot k^{-\gamma} + b < N$), $b \in [0,1]$ and c (where $c > 0$ and $g(k) < N$) until the degree distributions comply with the desired network size, average in coming and out going degrees and, minimal and maximal individual vertex degree. Figure 1 shows a few examples of power-law output distributions and the corresponding Poisson input distribution obtained in this way.

Once the desired input and output distributions are defined, we have to generate the network by connecting the vertices. Again, we defined a set of constraints:

- the input and output degree distributions are fixed, therefore, so is each node's input and output degree;
- multiple egdes where source and target vertices are the same are not allowed;

Where the first constraint is fairly straightforward and easy to satisfy, the second one may pose a problem, especially towards the end of the connection process. If one draws pairs at random, it will be increasingly difficult to find suitable couples as the process of connecting the nodes advances and more and more have reached their desired input and output connectivity. In fact, chances are that one gets stuck in the process with only unsuitable pairs, implying a fastidious rewiring task. We have come up with a scheme to tackle this problem and minimize the chances of having identical connections at the end of the process, thus rewiring. This algorithm works as follows:

1. we assign to each vertex v_i an input degree k^{in}_i and an output degree k^{out}_i according to the network's input, respectively output, degree distributions;

2. we built two lists of vertices, one for source nodes l_{out} and one for target nodes l_{in}, where each vertex will appear k_i^{in} times in l_{in} and k_i^{out} times in l_{out}. The size L of both lists will be the same and $L = N \cdot \overline{k}$;
3. we shuffle each list separately;
4. we try to connect elements of l_{in} to the corresponding element in l_{out} pairwise, starting at the top of the lists and working our way down the lists. If we hit a conflicting pair (both elements are the same vertex or the elements are already connected):
   (a) we shuffle separately both sublists going from the current position to the bottom;
   (b) if the conflict persists, we go back to (a). If the number of vertices to pair is down to one or if the conflict persists for more than a few tries, we disconnect all pairs and go back to (2);

This algorithm ensures that all the constraints are respected and is especially well suited to scale-free networks, where the output degree distribution guarantees that a vast majority of the vertices will only appear k_{min}^{out} times in the l_{out} list, thus limiting the probability of reaching a state where we have to restart the algorithm.

Discrete Timing of Events

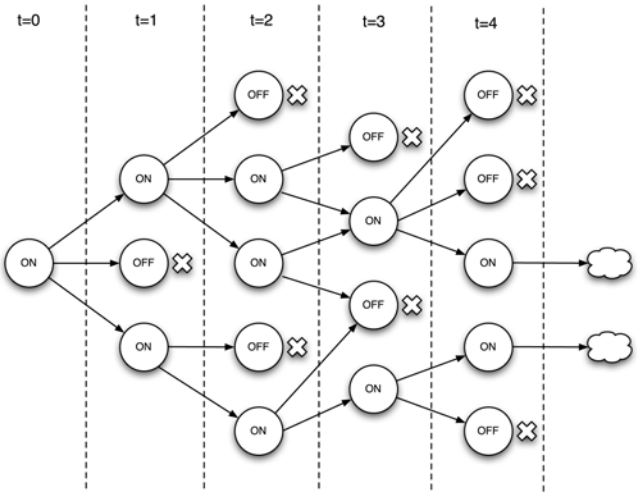
As we have seen above, in genetic regulatory networks, the expression of a gene depends on some transcription factors, whose synthesis does not appear to be neither fully synchronous nor instantaneous. Moreover, in some cases like the gene regulatory network controlling embryonic specification in the sea urchin (Davidson, 2002; Olivieri & Davidson, 2004), the presence of an activation sequence of genes can be clearly seen. We concluded that neither fully synchronous nor completely random asynchronous network dynamics are suitable models. In our opinion, the activation/update sequence in a RBN should be in some way related to the topology of the network.

In (Giacobini, Tomassini, De Los Rios & Pestelacci, 2006) we have proposed a topology-driven semi-synchronous update method, called *Cascade Update* (CU). Although not a faithful model for true biological gene activation sequences, we do believe that our proposed scheme is closer to biological reality than the previously proposed ones: fully synchronous (SU) and various asynchronous policies. Preliminary investigations on the behavior of this semi-synchronous update were encouraging but full analysis of its dynamics showed CU to act equivalently to the fully synchronous update scheme. In fact, after only a few time steps virtually all the nodes had to be updated at each time step, thus making the system essentially fully synchronous.

Aiming at remaining as faithful as possible to biologically plausible timing of events, we considered the influence of one node on another as active biological *activating* or *repressing* factors: only when the state of the node is turned or stays *on* has this node an effect on the subsequent nodes in the cascade. In contrary, nodes changing their state to or remaining *off* have no impact on nodes they are linked to, thus breaking the cascade. In other words, only the activation of an activator or a repressor will have a repercussion on the list of nodes to be updated at the next time-step. We have called this update scheme the Activated Cascade Update (ACU) (Darabos, Giacobini & Tomassini, 2007).

As a consequence of this semi-asynchronous update, the definition of point or cyclic attractors changes slightly, because the state of a network at any give time t is, from now on, not only determined by the

Figure 2. A possible activated cascade update sequence. At time $t = 0$ a node N_0 is chosen at random and updated according to it inputs, if the new state of N_0 is inactive, another starting node N_0 is chosen at random. At time $t = 1$ all the node receiving an input from N_0 are updated according to their own inputs, those becoming or remaining active (state on) decide which node will be updated at the next time step. The cascade continues according to this scheme.



individual state $s_i^t \in \{on, off\}$ of each node but also by the list of nodes to be updated at the next time step l_{t+1}. The concept of loose attractor has, in this context, no relevance. An interesting phenomenon is the emergence of a new type of attractors periodic attractors that cycle through all the right network configurations in the right order, but have nevertheless slightly different lists l_{t+1}. This possible as the fact that a node is updated does not automatically mean a change in its state at the next time step, thus no further change in the cascade. We call these configuration cyclic attractors (CCA) in contrast regular cyclic attractors as defined above.

METHODOLOGY AND SIMULATIONS

Using the MCM model described in Section 3.1 we produce networks having a scale-free distribution of the output degrees and a Poisson distribution of the input degrees. In this work we investigate the effect of the new ACU update scheme presented in Section 3.2 vs. the previous SU for a set of γ exponents of the scale-free distribution $\gamma \in \{2.0, 2.5, 3.0\}$. In an effort to probe the network scaling properties, we have simulated ensembles of graphs with $N \in \{100, 150, 200\}$, all with a connectivity of $\overline{k} = 4$. The results will be compared to classical RBN. Networks of all three sizes above are evolved and, in order to explore their behavior in three different regimes, we propose to vary $\overline{k} \in \{1.5, 2.0, 2.5\}$, thus keeping the probability p of the node update functions to $p = 0.5$.

For any given network produced, an update function is attributed to each node, consisting of a randomly initialized lookup table, the entries of which are determined by the input degree of that particular

node. Subsequently, we fix an initial configuration (IC), which is a set of initial boolean values chosen at random with probability $p = 0.5$, for each node of the network. Then, we let the system stabilize over a number of initial steps depending on the size N of the network (10'000 for $N = 100$, 20'000 for $N = 150$, and 30'000 for $N = 200$). During these preliminary steps the chosen update scheme determines the next nodes to be updated. This allows the system to evolve and maybe reach the basin of an attractor. Should the cascade stop because there is no node to be updated, another initial node is selected at random and the process is restarted with the same realization. After this transient period, we determine over another 1'000 time steps if the system has reached an attractor. If so, we define the length of that attractor as the minimum number of steps necessary to cycle through the attractor's configuration. This is repeated 20 times, each of them with a new set of update rules for each node. A particular topology, together with a given set of node functions, is called a realization. Each realization is exposed to 500 ICs. In order to be thorough, we study 50 different networks (i.e. 1'000 realizations) both random graph and scale-free topologies, each using the two update schemes.

Analysis of the Results

For the ensemble of these evolution we monitored the probability $p(m)$ for a network realization to have exactly m different attractors, and the probability $p(l)$ of an attractor to contain l different states.

Number of Attractors

During the simulations, we have analyzed for each IC of each realization whether the system has relaxed to a single state (point attractor) or cycled through the configurations of a periodic attractor. Biologically speaking, a point attractor has a very limited significance, because it would either mean that the system vegetates with no chance of evolving or adapting, and ultimately the death of the system, or identify the end of the differentiation cycle of a stem cell. Therefore, in Table 1 we show the total number of attractors found. We only consider attractors of length between 1 and 50 states.

Table 1. Number of attractors for synchronous (SU) and semi-synchronous (ACU) update schemes. Results are shown for scale-free GBN (SFBN) with all values of γ and for classical RBN for all values of $\overline{k}$, for all values of N

		$N = 100$		$N = 150$		$N = 200$	
		SU	ACU	SU	ACU	SU	ACU
SFBN	$\gamma = 2.0$	8	126	0	2	0	0
	$\gamma = 2.5$	27	379	0	3	0	0
	$\gamma = 3.0$	32	315	2	9	0	0
RBN	$\overline{k} = 1.5$	8'523	499'983	12'628	499'994	13'478	499'985
	$\overline{k} = 2.0$	7'086	497'763	7'193	499'933	6'200	499'878
	$\overline{k} = 2.5$	1'009	352'449	229	431'182	72	462'932

Table 2. Number of point attractors / cyclic attractors for synchronous (SU) and semi-synchronous (ACU) update schemes for networks with $N = 100$ nodes. Results are shown for scale-free GBN (SFBN) with all values of γ and for classical RBN for all values of $\overline{k}$, for $N = 100$. The average usage columns show how many times, in average, each attractor has been found.

		Number of attractors		Average occurrence	
		SU	ACU	SU	ACU
SFBN	$\gamma = 2.0$	0/8	105/21	1.12	1.36
	$\gamma = 2.5$	9/18	340/39	3.66	1.79
	$\gamma = 3.0$	4/28	266/50	6.875	3.212
RBN	$\overline{k} = 1.5$	795/7728	498297/1686	57.0406	1.00
	$\overline{k} = 2.0$	376/6710	451451/46312	46.9009	1.0043
	$\overline{k} = 2.5$	108/901	254261/98188	50.4896	1.347

We note that the number of attractors found with the ACU is greater than those found with SU. But in the case of SFBN, the table also shows there is a significant drop in the number of attractors in scale-free networks as N grows, and virtually no attractors at all for values of $N = 150$ and $N = 200$. In the case of ACU, we see that the systems still find a few attractors in cases where SU struggles.

In the case of classical RBNs, we observe that the number of attractors does not seem to be impacted by the scaling, and their number remains several orders of magnitude above that of scale-free structures. It fact, their number increases as N grows, and using ACU almost every IC of every realization leads to an attractor. On the contrary, under synchronous updating the overall number of attractors decreases drastically as the system is going from less to more chaotic.

As for the type of attractors found in the case of semi-synchronous update, over 90% are regular cyclic attractors for scale-free structures and over 80% for random topologies.

Table 2 shows detailed experimental results for networks with $N = 100$, the class of networks that offers most diversity in attractors. Here, we see the number of point attractors found with SFBN seems

Figure 3. Scale-free boolean networks. Fraction of realizations having a given number of attractors with (a) $\gamma = 2.0$ and (b) $\gamma = 3.0$ for $N = 100$

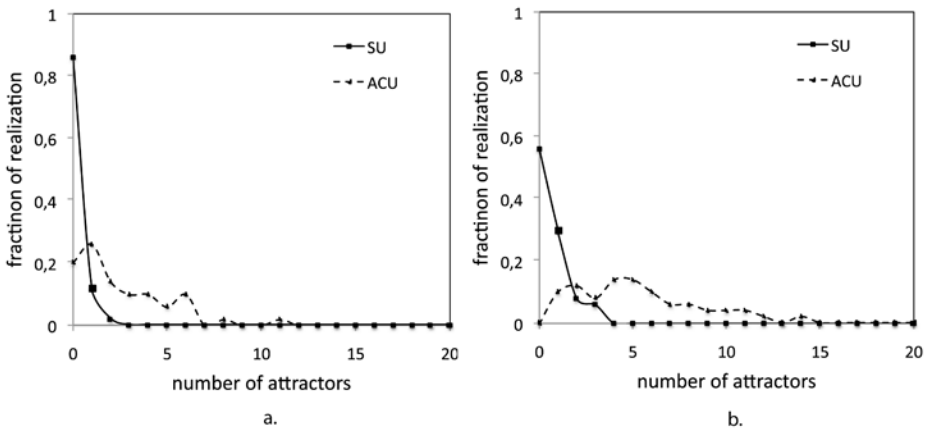
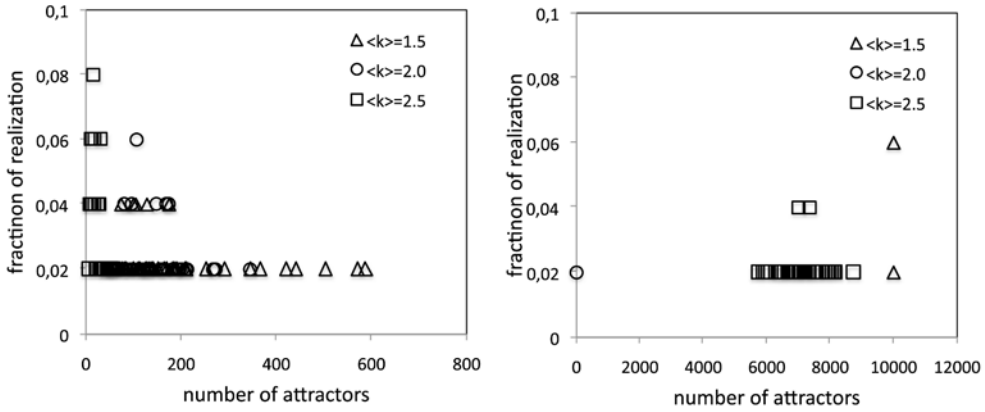


Figure 4. Classical random boolean networks. Fraction of realizations having a given number of attractors using (a) SU and (b) ACU for $N = 100$



to decrease, and it is the opposite with longer attractors. We also note that the number of occurrences of all attractors found in SFBNs increases slightly as the γ factor increases, but remains very low for both updates. Whereas this is also true for RBNs under ACU, it is not the case when the update is synchronous, where each attractor is found in average about 50 times.

Figures 4 and 5 compare the distribution of the number of attractors as a fraction of the different network realization respectively for scale-free networks and random networks of size $N = 100$.

Figure 3 compares the distribution of the number of attractors for the two extreme values of γ and the two different update mechanisms. There is a striking difference between the distribution using SU and ACU. In the synchronous case, most realizations have no attractors at all, whereas in the semi-synchronous case, the repartition is more even, with no realization having no attractor at all for $\gamma = 2.0$ and about 20% for $\gamma = 3.0$.

Figure 5. Scale-free boolean networks. Number of attractors found having a given length (between 1 and 50) with (a) $\gamma = 2.5$ *for (b)* $\gamma = 3.0$ *for* $N = 100$*. For ACU, over 90% of the attractors found are point attractors (not plotted here for readability reasons).*

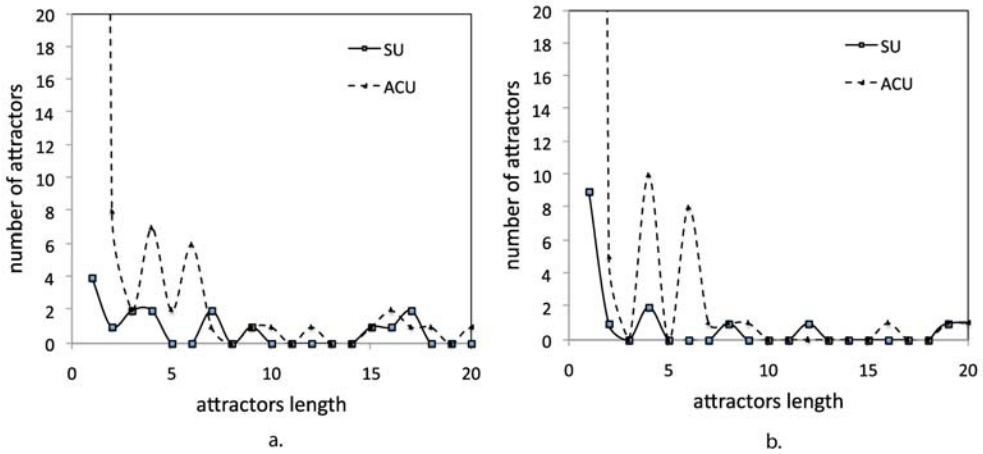


Figure 4 shows the attractors' number distribution on random topologies. Each figure contains all different values of $\overline{k}$, thus systems in different regimes, and (a) depicts the results for SU and (b) for ACU. Again, the contrast between the two update schemes is dramatic. Synchronous update leads to much less attractors than semi-synchronous ones, between 0 and 600 for SU and between 5'500 and 10'000 for ACU. In addition, we witness a segregation in the distribution of networks in different regimes. Systems in the chaotic regime tend to have in general fewer attractors than networks in the ordered regime or on the so called "edge of chaos", a well known phenomenon (Kauffmann, 1993). This cut is not as clear in the case of SFBNs.

Attractors Length

We have also computed the average length of cycles for attractor sizes 2 to 50. Considering their prominent number, we have omitted point attractor that artificially lower that average. Table 3 depicts the different average length of the attractors for the same network realizations shown in the previous section. We see that, although the number of cyclic attractors between the distinct cases sometimes differs by several orders of magnitude, the average attractors length within a network kind through the different regimes, though constantly higher for SU.

When comparing SFBNs and classical RBNs at different regimes, we note that the average length of the attractors obtained on scale-free structures is roughly twice as long as those derived from random topologies. This is astonishing when we compare the number of attractors found. Therefore, Figures 7 and 8 show the distribution of attractors for lengths from 1 to 50, over all the attractors. For readability reasons, we have limited the view of the number of attractors to 20 in the case of SFBNs, where the number of point attractors reaches a few hundreds and to 1500 for RBNs, where there are several thousands of single configuration attractor.

By looking at Figure 5 we note that the numer of attractors longer than 1 is greater when ACU is used, especially below length of 10. In addition, the number of attractors degreases with their length, this phenomenon is not as apparent with random structures in Figure 6, although, more so with ACU.

Table 3. Average length of cyclic attractors for synchronous (SU) and semi-synchronous (ACU) update schemes and their standard deviation in subscript. Results are shown for scale-free GBN (SFBN) with all values of γ and for classical RBN with all values of $\overline{k}$, for all values of N .

		$N = 100$		$N = 150$		$N = 200$	
		SU	ACU	SU	ACU	SU	ACU
SFBN	$\gamma = 2.0$	$23.87_{12.10}$	$11.90_{15.45}$	-	-	-	-
	$\gamma = 2.5$	$28.66_{16.80}$	$12.46_{12.48}$	-	4_{0}	-	-
	$\gamma = 3.0$	$23.14_{14.65}$	$15.64_{14.961}$	-	-	-	-
RBN	$\overline{k} = 1.5$	$13.76_{9.85}$	$4.48_{1.05}$	$16.65_{12.99}$	$4.15_{0.93}$	$17.21_{12.69}$	$4.20_{0.87}$
	$\overline{k} = 2.0$	$13.23_{10.25}$	$5.96_{4.35}$	$14.93_{10.98}$	$6.06_{4.36}$	$16.73_{11.39}$	$5.96_{4.31}$
	$\overline{k} = 2.5$	$14.36_{12.17}$	$9.58_{8.95}$	$12.29_{10.96}$	$9.97_{9.01}$	$12.68_{9.42}$	$9.94_{9.04}$

In all cases, we notice unexpected increase in value for attractors of even-length attractors. We can imagine that this is due to the binary nature of our networks. In the case of a point attractor, no node changes and there is thus a unique state. If we have 2 states, that could be explained by one node flipping from *on* to *off* and back at each time step. Similarly, 4 states would mean 2 nodes are flipping, and so on.

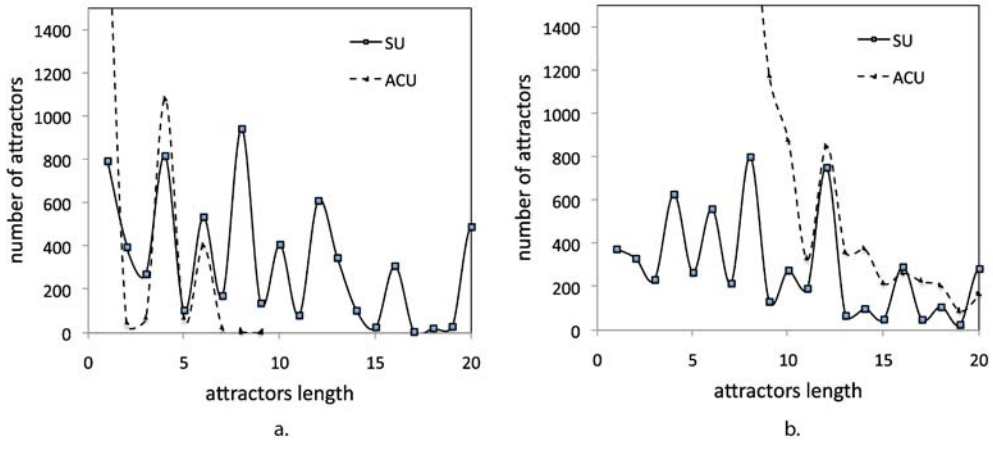
FAULT TOLERANCE OF RANDOM BOOLEAN NETWORKS

Failures in systems can occur in various ways, and the probability of some kind of error increases dramatically with the complexity of the systems. They can range from a one-time wrong output to a complete breakdown and can be system-related or due to external factors. Living organisms are robust to a great variety of genetic changes, and since RBN are simple models of the dynamics of biological interactions, it is interesting and legitimate to ask questions about their fault tolerance aspects.

Kauffman defines a one type of perturbation to RBN as "gene damage'' (Kauffman, 2000), that is the transient reversal of a single gene in the network. These temporary changes in the expression of a gene are extremely common in the normal development of an organisme. The effect of a single hormone can transiently modify the activity of a gene, resulting in a growing cascade of alternations in the expression of genes influencing each other. This is believed to be at the origine of the cell differentiation process and guides the development.

The effect of a gene damage can be measured by the size of the avalanche resulting from that single gene changing its behavior from active to inactive or vice-versa. The size of an avalanche is defined as the number of genes that have changed their own behavior at least once after the perturbation happened. Naturally, this change of behavior is compared to an unperturbed version of the system that would be

Figure 6. Random boolean networks. Number of attractors found having a given length (between 1 and 50) with (a) $\overline{k} = 1.5$ for (b) $\overline{k} = 2.0$ for $N = 100$. For ACU, over 90% of the attractors found are point attractors (not plotted here for readability reasons).



running in parallel. The size of the avalanche is directly related to the regime in which the RBN is; in the ordered regime, the cascades tend to be significantly smaller than in the chaotic regime. In real cells, where the regime is believed to lie on the edge of chaos, the cascades tend to be small too. Moreover, the distribution of the avalanche sizes in the ordered regime follows a power law curve (Kauffman, 2000), with many small and few large avalanches. In the chaotic regime, in addition to the power law distribution, 30-50 precent of the avalanches are huge. The distribution of avalanches size of RBNs in the ordered regime roughly fits the expectations of biologists, where most of the genes, if perturbed, are only capable of initiating a very small avalanche, if any. Fewer genes could cause bigger cascades, and only a handful can unleash massive ones.

Another measurement of the effect of transient gene reversal, is to compare the change in the configuration of the RBN between two consecutive time steps on an unperturbed system and on one where a single gene has been perturbed. The difference between two consecutive states s_t and s_{t+1} of the system is measured in therms of Hamming distance, that is the number of genes that have changed their expression between s_t and s_{t+1}, normalized over the network size.

Naturally, one can imagine more sophisticated failure schemes on models of genetic regulatory networks such as RBN. These failures are usually inspired by real biological experiments conducted on real organismes. For example the gene knock-out experiments measures the expression level of all genes, in cells wich a knock-out gene and in normal cells, using cDNA microarray data. In (Serra, Villani & Semeria, 2004; Serra, Villani, Graudenzi & Kauffman, 2007), Serra used this type of failure on RBNs to predict the size of real avalanches on microarray data. He showed that a very simple model with few inputs and random topologies can approximate the distribution of perturbation in gene expression levels with respect to microarray data. Moreover, he present a theoretical study showing that this simple model is actually valid in a particular type of network topologies.

Another notable perturbation inspired by real biological regulatory networks applied to RBNs is the gene duplication phenomenon suggested in (Aldana, Balleza, Kauffman & Resendiz, 2006). Aldana studies the robustness of genetic regulatory networks using RBNs and explore their behavior when exposed to nature-inspired genetic perturbations: gene duplications. He shows that an intrinsic property of such networks is to tend to preserve and multiply previous phenotypes, encoded in the attractor landscape of the network.

This section only offers a flavor of a vast aspect of the study of GBNs. The influence of spatial and temporal choices on the robustness of GBNs deserves a whole chapter of its own. We encourage readers interested in this matter to consider specialized literature.

TO MAKE A LONG STORY SHORT

Random Boolean Networks (RBN) have been introduced by Kauffman as a highly simplified model of genetic regulatory networks (Kauffmann, 1993). This extremely simple and abstract model has been studied in detail by analysis and by computer simulations of statistical ensembles of networks and it has been shown to be capable of extremely interesting dynamical behavior. First of all, it has been found that, as some parameters are varied such as the network's connectivity K, or the probability p of expressing a gene, i.e. of switching on the corresponding node's state, the RBN can go through a phase transition. Indeed, for every value of p, there is a critical value of connectivity $K_c(p)$ such that for values of K

below this critical value the system is in the ordered regime, while for values of K above this limit the system is said to be in the chaotic regime. Kauffman's suggestion is that cell types correspond to attractors in the RBN phase space, and only those attractors that are short and stable under perturbations will be of biological interest. Thus, according to Kauffman, RBN lying at the edge between the ordered phase and the chaotic phase can be seen as abstract models of genetic regulatory networks.

The original view of Kauffman, namely that these models may be useful for understanding real-life cell regulatory networks, is still valid, provided that the model is updated to take into account present knowledge about the topology of real gene regulatory networks, and the timing of events, without loosing its attractive simplicity.

From the structural and topological point of view, random networks with fixed connectivity degree K were a logical generic choice in the beginning, since the exact couplings in networks were generally unknown. Today it is more open to criticism since it does not correspond to what we know about the topology of biological networks. According to present data, many biological networks, including genetic regulatory networks, seem, in fact, to be of the scale-free type or hierarchical and not random as suggested, among others, by Albert and coworkers (Albert, 2005; Christensen, Gupta, Maranas, Albert, 2007). In addition, Aldana has analyzed of Boolean networks with scale-free topology. He has been able to define a phase space diagram for Boolean networks, including the phase transition from ordered to chaotic dynamics, as a function of the power law exponent.

From the point of view of the timing of events, standard RBN update their state synchronously. This assumption simplifies the analysis, but it is open to discussion when dealing with biologically plausible networks. In particular, for genetic regulatory networks, this is certainly not the case, as many recent experimental observations tend to prove. Rather, genes seem to be expressed in different parts of the network at different times, according to a strict sequence which depends on the particular network under study. The expression of a gene depends on several transcription factors, the synthesis of which appear to be neither fully synchronous nor instantaneous. Moreover, in some cases like the gene regulatory network controlling embryonic specification in the sea urchin, we can clearly see the presence of an activation sequence of genes.

In view of the above shortcomings of RBN as an abstract description of genetic regulatory networks we conclude that neither fully synchronous nor completely random asynchronous network dynamics are suitable models. Therefore, we have recently proposed a new, more biologically plausible model. It assumes a scale-free topology of the networks and we define a suitable semi-synchronous dynamics that better captures the presence of an activation sequence of genes linked to the topological properties of the network.

By computer simulations of statistical ensembles of networks, we have monitored the probability $p(m)$ for a network realization to have exactly m different attractors, and the probability $p(l)$ of an attractor to contain l different states. We noted that the number of attractors found with the ACU is greater than those found with SU. But in the case of SFBN, there is a significant drop in the number of attractors in scale-free networks as the number of nodes N grows. In the case of classical RBNs, we observe that the number of attractors does not seem to be impacted by the scaling, and their number remains several orders of magnitude above that of scale-free structures.

We have also computed the average length of cycles for attractor sizes 2 to 50. We see that, although the number of cyclic attractors between the distinct cases sometimes differs by several orders of magnitude, the average attractors length within a network kind through the different regimes, though constantly

higher for SU. When comparing SFBNs and classical RBNs at different regimes, we note that the average length of the attractors obtained on scale-free structures is roughly twice as long as those derived from random topologies. This is astonishing when we compare the number of attractors found.

From a biological point of view, RBNs in general and their subsequent evolutions, help understanding the dynamics of the complex biological systems, such as GRNs. Moreover, without giving up their attractive simplicity, Boolean Network model refinements in any aspect, spacial with new network topologies, temporal with more realistic update sequencing schemes or other, can be used in reverse engineering modeling techniques to unveil new interactions among components of biological regulatory networks.

ACKNOWLEDGMENT

The authors thank F. Di Cunto and P. Provero of the University of Torino (Italy) for the useful discussions and suggestions on biological regulatory networks. M. Tomassini and Ch. Darabos gratefully acknowledge financial support by the Swiss National Science Foundation under contract 200021-107419/1. M. Giacobini acknowledge funding (60% grant) by the Ministero dell'Universitá e della Ricerca Scientifica e Tecnologica.

REFERENCES

Albert, R. (2004). Boolean modeling of genetic regulatory networks. In E. Ben-Naim, H. Frauenfelder & Z. Toroczkai (Eds.), *Complex networks.* (LNP, pp. 459-479). Berlin: Springer.

Albert, R. (2005). Scale-free networks in cell biology. *Journal of Cell Science, 118*, 4947–4957. doi:10.1242/jcs.02714

Albert, R., & Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics, 74*, 47–97. doi:10.1103/RevModPhys.74.47

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D. Nonlinear Phenomena, 185*, 45–66. doi:10.1016/S0167-2789(03)00174-X

Aldana, M., Balleza, E., Kauffman, S. A., & Resendiz, O. (2006). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology, 245*, 433–448. doi:10.1016/j.jtbi.2006.10.027

Aldana, M., Coppersmith, S., & Kadanoff, L. P. (2003). Boolean dynamics with random couplings. In E. Kaplan, J. E. Marsden & K. R. Sreenivasan (Eds.), *Perspectives and problems in nonlinear science.* Springer Applied Mathematical Sciences Series (pp. 23-89). Berlin: Springer.

Christensen, C., Gupta, A., Maranas, C. D., & Albert, R. (2007). Inference and graph-theoretical analysis of Bacillus Subtilis gene regulatory networks. *Physica A, 373*, 796–810. doi:10.1016/j.physa.2006.04.118

Darabos, C., Giacobini, M., & Tomassini, M. (2007). Semi-synchronous activation in scale-free Boolean networks. In F. Almeida, E. Costa, et al. (Eds.), *Advances in Artificial Life, 9th European Conference, ECAL2007* (LNAI, pp. 976-985), Heidelberg. Springer-Verlag.

Davidson, E. H. (2002). A genomic regulatory network for development. *Science, 295,* 1669–1678. doi:10.1126/science.1069883

Derrida, B., & Pomeau, Y. (1986). Random networks of automata: A simple annealed approximation. *Europhysics Letters, 1*(2), 45–49. doi:10.1209/0295-5075/1/2/001

Edwards, R., & Glass, L. (2006). A calculus for relating the dynamics and structure of complex biological networks. In R. S. Berry & J. Jortner (Eds.), *Advances in chemical physics* (pp. 151-178). New York: J. Wiley and Sons.

Gershenson, C. (2004). Updating schemes in random Boolean networks: Do they really matter? In J. Pollack (Ed.), *Artificial Life IX Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (pp. 238-243). MIT Press.

Giacobini, M., Tomassini, M., De Los Rios, P., & Pestelacci, E. (2006). Dynamics of scale-free semi-synchronous Boolean networks. In L. M. Rocha, et al. (Eds.), *Artificial Life X* (pp. 1-7). Cambridge, MA: The MIT Press.

Harvey, I., & Bossomaier, T. (1997). Time out of joint: Attractors in asynchronous random boolean networks. In P. Husbands & I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life* (pp. 67-75). Cambridge, MA: The MIT Press.

Hassoun, M. H. (1995). *Fundamentals of artificial neural networks*. Cambridge, MA: MIT Press.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology, 22,* 437–467. doi:10.1016/0022-5193(69)90015-0

Kauffman, S. A. (1993). *The origins of order*. New York: Oxford University Press.

Kauffman, S. A. (2000). *Investigations*. New York: Oxford University Press.

Mesot, B., & Teuscher, C. (2003). Critical values in asynchronous random boolean networks. In W. Banzhaf (Ed.), *Advances in Artificial Life, ECAL2003.* (LNAI, pp. 367-376). Berlin: Springer.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review, 45,* 167–256. doi:10.1137/S003614450342480

Olivieri, P., & Davidson, E. H. (2004). Genes regulatory network controling embryonic specification in the sea urchin. *Current Opinion in Genetics & Development, 14,* 351–360. doi:10.1016/j.gde.2004.06.004

Oosawa, C., & Savageau, M. A. (2002). Effects of alternative connectivity on behavior of randomly constructed Boolean networks. *Physica D. Nonlinear Phenomena, 170,* 143–161. doi:10.1016/S0167-2789(02)00530-4

Serra, R., Villani, M., Graudenzi, A., & Kauffman, S. A. (2007). Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. *Journal of Theoretical Biology, 246,* 449–460. doi:10.1016/j.jtbi.2007.01.012

Serra, R., Villani, M., & Semeria, A. (2004). Genetic network models and statistical properties of gene expression data in knock-out experiments. *Journal of Theoretical Biology, 227,* 149–157. doi:10.1016/j.jtbi.2003.10.018

Tomassini, M., Giacobini, M., & Darabos, C. (2007). Performance and robustness of cellular automata computation on irregular networks. *Advances in Complex Systems*, *10*, 85–110. doi:10.1142/S0219525907001124

Vázquez, A., Dobrin, R., Sergi, D., Eckmann, J.-P., Oltvai, Z. N., & Barabàsi, A.-L. (2004). The topological relationships between the large-scale attributes and local interactions patterns of complex networks. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(52), 17940–17945. doi:10.1073/pnas.0406024101

Section 6

Heterogenous Data

Chapter 19
A Linear Programming Framework for Inferring Gene Regulatory Networks by Integrating Heterogeneous Data

Yong Wang
Academy of Mathematics and Systems Science, China

Rui-Sheng Wang
Renmin University, China

Trupti Joshi
University of Missouri, USA

Dong Xu
University of Missouri, USA

Xiang-Sun Zhang
Academy of Mathematics and Systems Science, China

Luonan Chen
Osaka Sangyo University, Japan

Yu Xia
Boston University, USA

ABSTRACT

There exist many heterogeneous data sources that are closely related to gene regulatory networks. These data sources provide rich information for depicting complex biological processes at different levels and from different aspects. Here, we introduce a linear programming framework to infer the gene regulatory networks. Within this framework, we extensively integrate the available information derived from multiple time-course expression datasets, ChIP-chip data, regulatory motif-binding patterns, protein-protein

DOI: 10.4018/978-1-60566-685-3.ch019

interaction data, protein-small molecule interaction data, and documented regulatory relationships in literature and databases. Results on synthetic and real experimental data both demonstrate that the linear programming framework allows us to recover gene regulations in a more robust and reliable manner.

INTRODUCTION

Cells efficiently carry out molecular synthesis, energy transduction, and signal processing across a range of environmental conditions by gene networks, which we define broadly as networks of interacting genes, proteins, and metabolites. Microarray technologies enable the simultaneous measurement of all RNA transcripts in a cell, producing tremendous amounts of gene expression data from different research groups. For instance, the Stanford Microarray Database (SMD) has deposited data for 70,113 experiments, from 341 labs and 56 organisms, as of 2007 (Demeter et al., 2007). Thus there is a pressing need for the development of sophisticated algorithms for reverse-engineering gene networks. So far, many computational algorithms have been developed to analyze gene expression profiles to detect dependencies among genes over different conditions.

Generally speaking, there are two strategies for studying the relationships among genes. The "physical (direct) interaction" approach seeks to identify true physical interactions between regulatory proteins and their binding promoters to reconstruct the so-called transcriptional regulatory network (R. S. Wang, Wang, Zhang, & Chen, 2007). The second strategy, the "genetic (indirect) interaction" approach seeks to identify regulatory influences between RNA transcripts to reconstruct the so-called gene regulatory network (Y. Wang, Joshi, Zhang, Xu, & Chen, 2006). Thus, in general, the regulator transcripts may exert their effects indirectly through the action of proteins, non-coding RNA, metabolites, and the cell environmental factors. An advantage of the influence strategy is that the model can implicitly capture regulatory mechanisms at the protein and metabolite level that are not physically measured (Gardner & Faith, 2005). In this study we focus on the inference problem for gene regulatory networks. The detailed descriptions on the first strategy, i.e. inferring transcriptional regulatory networks, can be found in (R. S. Wang et al., 2007).

So far, a wide variety of approaches have been proposed to infer gene regulatory networks from time-course data or perturbation experiments (De Hoon, Imoto, Kobayashi, Ogasawara, & Miyano, 2003; Dewey & Galas, 2001; Friedman, 2004; Gardner, di Bernardo, Lorenz, & Collins, 2003; Holter, Maritan, Cieplak, Fedoroff, & Banavar, 2001; Husmeier, 2003; Nachman, Regev, & Friedman, 2004; Tegner, Yeung, Hasty, & Collins, 2003). These approaches include discrete models of Boolean networks and Bayesian networks, and continuous models of neural networks and difference/differential equations. A common challenge for all these models is the scarcity of the data, since a typical gene expression dataset consists of relatively few time points (often less than 20) with respect to a large number of genes (generally over thousands). In other words, the number of genes far exceeds the number of time points for which data are available, making the problem of determining gene regulatory network structure a difficult and ill-posed one (D'Haeseleer, Liang, & Somogyi, 2000).

On the other hand, there are many heterogeneous data sources closely related to gene regulatory networks. These data sources provide rich information for depicting complex biological processes in cellular systems at different levels and from different aspects. It is necessary and important to understand gene expression and regulation through mining these data sources. Currently high-throughput microar-

ray technologies have produced tremendous amounts of gene expression data from different labs. At the same time, a large amount of protein-based data exist such as ChIP-chip, protein-protein interaction, and protein-small molecule interaction, which can also provide valuable information. Even though each experiment provides only limited information, these data are increasingly accumulated over many species and can be freely accessed from public databases and individual websites. It is therefore valuable and challenging to integrate gene expression data with other protein-based data generated by different research groups. If such large amounts of data from different experiments or conditions are combined and further exploited in an integrative and systematic manner, the scarcity of data can be greatly alleviated and the more accurate reconstruction of gene regulatory networks can be expected.

To address these challenges, we proposed a novel method to combine multiple time-course microarray datasets from different conditions for inferring gene regulatory networks (Y. Wang, Joshi, Zhang et al., 2006). The proposed method, called GNR (Gene Network Reconstruction tool), is based on linear-programming (LP) and a decomposition procedure. The method ensures the derivation of the network structure that is most consistent with all datasets. As a result, the method not only significantly alleviates the problem of data scarcity, but also markedly improves the prediction reliability. We tested GNR using both simulated data and experimental data in yeast and *Arabidopsis*. The result demonstrates the effectiveness of GNR in terms of predicting new gene regulatory relationships.

Different experimental technologies measure different aspects of a biological system, typically with different systematic biases. For example, current high-throughput assays are usually associated with high false-negative and false-positive rates. Thus, microarray data alone have a limited utility in inferring gene regulatory networks. From the viewpoint of systems biology, the integration of data from different sources provides an effective strategy to deal with this issue by reinforcing consistent and reliable observations and removing inconsistent and noisy ones. Moreover, because different experimental technologies provide different types of insights into a biological system, the integration of multiple data types offers the most comprehensive information about a particular cellular process (Hwang et al., 2005). For example, gene perturbation experiments (e.g., knockouts or RNA interference) may indicate relationships between genes due to direct or indirect genetic interactions. In contrast, chromatin immunoprecipitation chip data may reveal direct protein–DNA interactions or cofactor associations with bound transcription factors. Combining them together with microarray data provides a much more detailed view of the regulatory network than either alone.

In this chapter, we introduce a new computational strategy to infer gene regulatory networks based on linear programming. The main advantage of our strategy is to recover gene regulations in a robust and reliable manner by including all the available information derived from multiple expression datasets at different conditions and time points, motif-occurrence, ChIP-chip data, protein-protein interaction, protein-small molecule interaction, published literature and databases, and knockouts or RNA interference experiments. Furthermore, we can incorporate external inputs or perturbations such as small molecules into the formulation so that molecular targets (genes) can be identified in a systematic way.

The chapter is organized as follows: Firstly, the heterogeneous data sources for deriving gene regulatory relationships are briefly summarized. Secondly, we group the existing prior information into hard and soft constraints, describe the gene regulatory network by linear differential equations, and introduce a linear programming model to integrate data. Thirdly, both synthetic data and real experimental data are used to demonstrate the effectiveness and efficiency of our method. Finally, future research directions are discussed.

HETEROGENEOUS DATA SOURCES

Organisms use dynamic interactions of hundreds of genes to adapt to changes in the environment. To unravel this regulatory complexity, multiple technologies have been developed to detect the dependencies among genes, generating large amounts of heterogeneous data (Joyce & Palsson, 2006). These data depict the living cell from different aspects and angles. Here we give a brief summary of the existing data sources related to gene regulation relationships and their characteristics.

Multiple Time-Course Expression Data

DNA microarray experiments are usually classified based on the type of array used in the experiment (cDNA and oligonucleotide arrays) or according to the organism that is profiled. From the viewpoint of gene regulatory network modeling, we distinguish between static and time series experiments. In static expression experiments, a snapshot of the expression of genes in different samples is measured. In time series expression experiments, a temporal process is measured at various time intervals. Another important difference between these two types of data is that while static data from a sample population (e.g. ovarian cancer patients) are assumed to be independently and identically distributed, time series data exhibit a strong autocorrelation between successive points.

Since many biological systems are dynamic systems, temporal profiles of gene expression levels during a given biological process can often provide more insights into how gene expression levels evolve in time and how genes are dependent among each other during a given biological process. One important feature of such time-course gene expression data is the possible dependency of gene expression levels across time points for a given gene. In addition, as gene expression levels evolve over time, time intervals can be an important factor that affects the gene expression levels. Methods which can preserve the time sequence and the time dependence of the observed data are needed for analyzing the time-course gene expression data.

Due to the limitation of experimental technologies, a typical single time-course gene expression dataset consists of relatively few time points (often less than 20). On the other hand, multiple gene expression data generated by different groups on many species are increasingly available and accessible from public databases or websites. By combining and exploiting such large amounts of data from different experiments or conditions in an integrative and systematic manner, we can expect a more accurate reconstruction of the gene regulatory networks. It is worth mentioning that simply arranging multiple time-course datasets into a single expression profile dataset is inappropriate due to data normalization issues and lack of temporal relationships among these datasets.

ChIP-Chip Data

Protein-DNA interactome data concerns the interactions between proteins and DNA, particularly between transcription factors and their target promoters. They fundamentally define the transcriptional regulatory network of the cell. The recently developed ChIP-chip methodology involves the chromatin immunoprecipitation of an epitope-tagged transcription factor (TF) bound to DNA fragments containing target promoters, followed by the hybridization of those amplified DNA fragments to an intergenic microarray. Currently large amounts of ChIP-chip data in yeast and other organisms are publicly available. For example, genome-wide location data performed in yeast by (Harbison et al., 2004; Lee et al.,

2002) contain information regarding the binding of 204 regulators to their respective target genes in rich medium, and can be downloaded from their websites (http://web.wi.mit.edu/young/regulatory_code/ and http://web.wi.mit.edu/young/regulatory_network/).

ChIP-chip data have the advantage that they provide a direct biochemical link between TFs and promoters and have the potential to identify targets without knowing the activating conditions. From this viewpoint, ChIP-chip data are a very important source of information for analyzing direct transcriptional regulatory interactions.

Regulatory Motif Occurrence Data

We can also use the genome sequence data to infer regulatory relationships by systematically analyzing gene upstream regions in the genome to identify potential regulatory elements (also known as regulatory binding motifs). These motifs, often represented as regular expressions, were transformed into the corresponding weight matrices. We can then simply count the occurrences of regular expression-type patterns with the goal of identifying possible gene regulatory relationships. The weight matrices corresponding to these motifs are subsequently used to screen all intergenic sequences. The higher the score of a motif hit in a gene, the more likely it will be a regulatory relationship (Brazma, Jonassen, Vilo, & Ukkonen, 1998).

Protein-Protein Interaction Data

Proteins are the products of gene transcription and translation, and they play important roles in a cell. Protein-protein interactions occur in many cellular processes, such as signaling cascades and enzyme-complex formation. Identifying all functional protein-protein interactions is important for understanding the structure and function of the integrated cellular network. Currently, a lot of experimental protein-protein interaction data are available on the web (http://www.thebiogrid.org/).

Protein-protein interaction data can be roughly classified into two classes: physical and genetic interactions. There are many methods for mapping physical and genetic interactions. From BioGRID (Breitkreutz et al., 2008), the physical methods include affinity capture MS, two-hybrid, affinity capture western, and reconstituted complex, whereas the genetic methods include synthetic lethality, synthetic growth defect, epistatic miniarray profile, dosage rescue, and phenotypic enhancement. Here we would like to illustrate in detail genetic interaction relationships. For example, synthetic lethality is a genetic phenomenon in which two non-lethal mutations yield a lethal phenotype when combined. This phenomenon signifies the existence of genetic interactions between the two affected genes. Hence, genetic interactions may overlap with direct physical interactions or indirect logical interactions between genes as shown by perturbation experiments (e.g., knockouts or RNA interference).

Protein-Small Molecule Interaction Data

Small molecules can be used to dissect diverse biological processes, such as cellular metabolism, signal transduction and intracellular protein trafficking (Alaoui-Ismaili, Lomedico, & Jindal, 2002). Recently, the proliferation of web-based chemical databases has made information about an increasing number of compound structures and their biological properties publicly available. Among these databases are ChemBank, ZINC, PubChem, ChemDB, ChemMine, ChEBI, and DrugBank. Small molecule and protein

binding data are also abundant. For example, the DrugBank database is a unique bioinformatics and cheminformatics resource that combines detailed drug data with comprehensive drug target information (Wishart et al., 2007). BindingDB currently contains 20,000 experimentally determined protein–ligand complexes from the literature and PDB (Liu, Lin, Wen, Jorissen, & Gilson, 2007). Binding MOAD is a database of 9,836 protein–ligand crystal structures (Benson et al., 2007). STITCH contains interactions for over 68,000 chemicals and over 1.5 million proteins in 373 species (Kuhn, von Mering, Campillos, Jensen, & Bork, 2008). These data provide useful information for the interactions between the gene regulatory network inside the cell and the environmental factors outside the cell.

Literature and Database Data

More reliable sources for gene regulatory relationships are from the literature and curated databases. For example, YEASTRACT (Yeast Search for Transcriptional Regulators And Consensus Tracking) is a curated repository of more than 12,500 regulatory associations between transcription factors and target genes in *Saccharomyces cerevisiae* (Teixeira et al., 2006), based on more than 900 bibliographic references. The information in YEASTRACT is updated regularly to match the recent literature on yeast regulatory networks. Since the regulatory relationships from literature and databases are usually generated by small-scale experiments, they are believed to be of high quality compared to large-scale experiments.

Co-Expression Relationships from Compendium Data

In addition to time-course data, the steady state gene expression data are also available in the databases. They can be assembled into gene expression profile or compendium data and used to extensively analyze the gene co-expression relationship. These microarray profile data are very useful in our derivation of gene regulatory network in two ways.

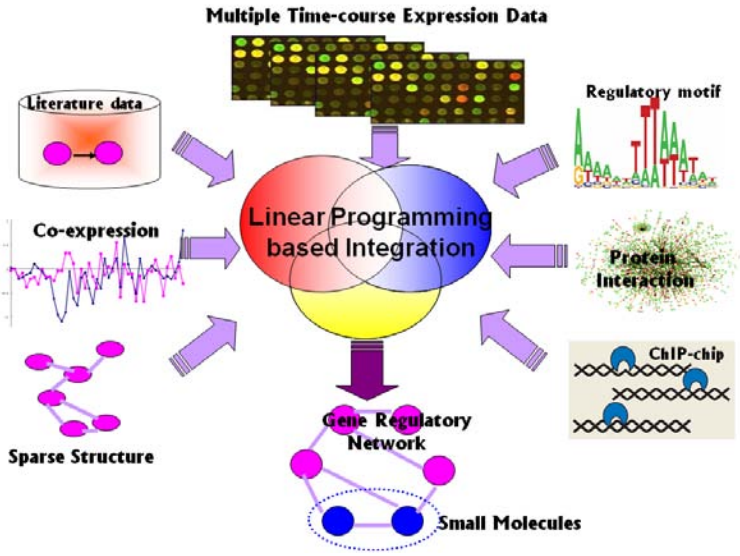
On one hand, gene expression data can be used to find co-expressed gene pairs (which display high correlation coefficient or mutual information score amongst different expression experiments). Over the past few years, several lines of evidence suggest that co-expressed genes possessing similar expression patterns across a set of steady states are likely to encode proteins that participate in the same metabolic pathway, form a common structural complex, or might be regulated by the same mechanism (Butte, Tamayo, Slonim, Golub, & Kohane, 2000). At the same time, diverse regulatory mechanisms may be responsible for the observed co-expression relationships.

On the other hand, gene expression data can be used to pick out the gene pairs which do not possess any co-expression relationships. That is to say, we can use large scale co-expression analysis in different conditions to reveal gene pairs which correlate weakly in terms of their expression level across various conditions. These identified pairs can be used as non-coregulatory samples approximately in our network inference model.

Prior Information about the Network Structure

In addition to various experimental data sources, we can also incorporate prior information about the network structure. For example, from the viewpoint of topology, it is commonly believed that gene regulatory network is sparse in nature, i. e. each gene is only genetically affected by a limited number

Figure 1. The graphic depiction of the strategy to integrate heterogeneous data using a linear programming framework



of genes (Gardner et al., 2003; Yeung, Tegner, & Collins, 2002). Furthermore, some people argue that the gene regulatory network possesses common properties of complex networks such as small world and scale free (Gustafsson, Hornquist, & Lombardi, 2005). It is straightforward to incorporate this prior information into our inference model. The main idea here is to make the gene regulatory network sparse so that it is biologically plausible. Such a strategy has been widely used (Gustafsson et al., 2005; Y. Wang, Joshi, Xu, Zhang, & Chen, 2006; Y. Wang, Joshi, Zhang et al., 2006; Yeung et al., 2002). For instance, a heuristic manipulation of sparseness is used in the procedure of the network reconstruction by computational analysis on a series of time points (Gardner et al., 2003; Yeung et al., 2002). A sparse scheme is performed by specifying the average number of connections for every gene in (Gardner et al., 2003). Another strategy is to use additional information from the microarray analysis and from the published literature to reduce the size of the problem and increase the reliability of the results (Nariai, Tamada, Imoto, & Miyano, 2005).

LINEAR PROGRAMMING FRAMEWORK FOR DATA INTEGRATION

Figure 1 illustrates the scheme of our proposed method. The time-course datasets of microarray experiments from different conditions or perturbations are collected. A gene regulatory network is described by ordinary differential equations (ODE). To infer the relationships between genes, the co-expression relations from time-course datasets and previously known regulations from the heterogeneous sources are collected as prior information, which are converted to hard and soft constraints respectively. In the end, the most consistent gene regulatory network is obtained with a linear programming-based algorithm.

Linear Differential Equations for Gene Regulatory Network

In general, a genetic network can be expressed by a set of nonlinear differential equations. Almost all of the existing approaches for gene regulatory network inference use linear or additive models, primarily due to the complex structures of biological systems and the scarcity of data (R. S. Wang et al., 2007; Y. Wang, Joshi, Xu et al., 2006; Y. Wang, Joshi, Zhang et al., 2006). Furthermore, linear equations can capture the main features of the network near the steady state, and can provide a good starting point for further modeling and analysis.

A common experimental technique for elucidating genetic network architecture is microarray measurements after different perturbations to the cell. An external perturbation means an experimental treatment that can alter the transcription rate of the genes in the cell. An example of perturbation is the alteration of the environment, treatment of the cell with a chemical compound, or genetic perturbation involving over- or under-expression of particular genes. Recent developments in large-scale genomic technologies enable researchers to measure gene expression profiles at multiple time points following perturbation of the genes of interest. We will extend the linear differential equation model to reconstruct gene regulatory networks and identify compound targets by considering the external perturbations outlined in this chapter. The model is based on relating the changes of gene transcript concentrations to each other and to the external perturbations.

Assume that there are N microarray datasets X^1, X^2 …, X^N with m_1, m_2, …, m_N time points respectively for one organism. These time-course datasets may be measured under various environments or stimuli by different labs. Let us first consider one time-course dataset with m time points. A linear differential equation can be used to represent the rate of synthesis of a transcript as a function of the concentrations of other transcripts in a cell and the external perturbations:

$$\frac{dx(t)}{dt} = Jx(t) + Pc(t), \qquad t = t_1, t_2, ..., t_m \tag{1}$$

where $\mathbf{x}(t) = (x_1(t), \ldots, x_n(t))^T \in R^n$, $x_i(t)$ is the expression level (mRNA concentrations) of gene i at time point t. $J=(J_{ij})_{n\times n}$ is an n×n connectivity matrix with elements J_{ij} representing the effect of gene j on gene i with a positive, zero, or negative sign, indicating activation, no interaction, and repression, respectively. $P=(P_{ij})_{n\times s}$ is an n×s matrix representing the effect of the s perturbations or s small molecules on x, and $c(t) \in R^s$ represents the external perturbations with s compounds at time t (In principle, the external perturbation can be of virtually any type. For example, an external environmental factor, a small molecule, an enzyme, a microRNA, or a post-translationally modified protein). A non-zero element P_{ij} of P implies that the i-th gene is a direct target of the j-th perturbation or compound. Identifying P is an important first step towards biological function discovery of small molecules and drug design.

We can rewrite Equation (1) in a compact form for all time points of one dataset by matrix notation:

$$\frac{d\mathbf{X}}{dt} = J\mathbf{X} + P\mathbf{C} \tag{2}$$

where $\mathbf{X}=(\mathbf{x}(t_1),...,\mathbf{x}(t_m))$ and $d\mathbf{X}/dt=(d\mathbf{x}(t_1)/dt,...,d\mathbf{x}(t_m)/dt)$ are n×n matrices with the first derivative of mRNA concentration $dx_i(t_j)/dt=[x_i(t_{j+1})-x_i(t_j)]/[t_{j+1}-t_j]$ for i=1,...,n; j=1,...,m. Although the forward differ-

ence approximation here is utilized for numerical computation of dx/dt, backward or other difference approximation methods can be applied similarly. Suppose that there are s external perturbation compounds, then $\mathbf{C}=(\mathbf{c}(t_1),...,\mathbf{c}(t_m))$ is an s×m matrix representing the s perturbations. The unknowns to be calculated are connectivity matrix J and P.

Equation (2) can be reformulated as:

$$\frac{d\mathbf{X}}{dt} = [J, P]\begin{bmatrix}\mathbf{X}\\ \mathbf{C}\end{bmatrix} \tag{3}$$

We then apply Singular Value Decomposition (SVD) to $[\mathbf{X}^T\mathbf{C}^T]$:

$$\begin{bmatrix}\mathbf{X}\\ \mathbf{C}\end{bmatrix}^T_{m\times(n+s)} = \mathrm{U}_{m\times(n+s)}\mathrm{S}_{(n+s)\times(n+s)}\mathrm{V}^T{}_{(n+s)\times(n+s)} \tag{4}$$

where U is a unitary m×(n+s) matrix of left eigenvectors, **S**=diag(s_1, …, s_{n+s}) is a diagonal (n+s)×(n+s) matrix containing the (n+s) eigenvalues, and V^T is the transpose of a unitary (n+s)×(n+s) matrix of right eigenvectors. We can then obtain a specific solution of each dataset with the smallest L_2 norm for the Jacobian matrices J and P:

$$[\overline{J}, \overline{P}] = \frac{dX}{dt} US^{-1}V^T \tag{5}$$

where S^{-1}= diag($1/s_1$, …, $1/s_{n+s}$) and $1/s_i$ is set to be zero if s_i=0.

Similarly, we can infer N networks from N datasets respectively:

$$[\overline{J^k}, \overline{P^k}] = \frac{dX^k}{dt} U^k S^{k-1}V^{kT} \tag{6}$$

where the superscript k=1,...,N is the index of the k-th dataset. Note that without explicit normalization, J^k for each dataset is already a normalized matrix for different experiments with different time intervals due to the form of Equation (5).

Thus, the general solution of the Jacobian matrix $J^k = (J^k{}_{ij})$ and $P^k = (P^k{}_{ij})$ for each dataset k is expressed by

$$[J^k, P^k] = [\overline{J^k}, \overline{P^k}] + Y^k V^{kT} \tag{7}$$

Equation (7) represents all possible networks that are consistent with each microarray dataset, depending on arbitrary variables Y^k. $Y^k=(Y^k{}_{ij})$ is an n×(n+s) matrix, where $Y^k{}_{ij}$ is zero if $s^k{}_j \neq 0$ and is otherwise an arbitrary bounded scalar coefficient, i.e., $|Y^k{}_{ij}| \leq M$, where M is a given positive constant. In the next subsection we will explain how to construct the most consistent gene regulatory network [J P] from all $[J^k\ P^k]$ by determining Y^k, k=1,…, N.

Hard and Soft Constraints

Before formally introducing the linear programming based integration framework, we briefly categorize the prior information. In our differential equation model we use the Jacobian matrix J to represent the gene regulatory relationships. The regulatory relationships can be directed, signed, and weighted. For example, element J_{ij} represents an effect of gene j on gene i, while J_{ji} represents an effect of gene i on gene j. Thus the influence between gene i and gene j is directed. Furthermore, a sign associated with J_{ij} represents a specific role of regulation. For example, if the sign of J_{ij} is positive, gene j is the activator of gene i. On the other hand, if the sign of J_{ij} is negative, gene j is the repressor of gene i. Furthermore the associated weight (the absolute value) of element J_{ij} indicates how strong the regulatory interaction is. Obviously, a zero weight of J_{ij} indicates no interaction between two genes.

Thus, existing prior information about regulatory relationships can be roughly classified as follows:

- **Undirected.** Given a gene pair, we only know that there is a regulatory interaction between them, but the information about regulator and target gene is unavailable. For example, protein-protein interactions occur at the protein level instead of gene level, and they provide us with some hints that there exist certain relationships between two genes but no directional information. The (non-) co-expression relationships also belong to this class.
- **Directed and un-signed.** In this class, we know that there is a directed regulatory interaction but we do not know if it is an activation or repression regulation. For example, the ChIP-chip data and regulatory motif occurrence data tell us about the transcriptional regulation relationship, i.e. a transcription factor binds to the promoter region of a target gene and possibly influences its expression level, but the activating or repressing information is not available.
- **Directed and signed.** In this class, we know more about the regulation, both the regulation direction and the activation or repression role. Literature and the existing databases provide such reliable information. Also we can derive such information from the GO functional annotations. For example, activation relation can be obtained by selecting those regulatory relations such that the regulator is either an activator or co-activator in GO function annotation. Similarly the set of repression relation can be obtained by selecting those regulatory relations such that the regulator is either a repressor or a co-repressor.

In practical implementation, we can simply treat the undirected relationship as two directed and un-signed relationships (for example, the undirected relationship between A and B can be decomposed into two directed relationships: A to B and B to A). After this treatment, there are essentially two kinds of prior information: directed signed and directed unsigned. Available information from heterogeneous sources can be incorporated into our linear programming framework as soft and hard constraints, depending on the certainty of the information. The hard constraints include the directed and signed relationships. Their signs must be guaranteed and weight should be inferred. The soft constraints include the directed and un-signed relationships and their signs and weights are determined in the integration process.

Let us compare our method with traditional machine learning methods in terms of prior information incorporation. From the viewpoint of machine learning, the reliable information (gold standard positive and negative data) should be treated in a supervised way, i.e. they are labeled as positive or negative samples which are used to train the classifier. In our model, hard constraints similarly ensure that such

reliable prior information (directed and signed) is properly learned. The difference is that our method ensures that the reliable prior information must appear in the final results while gold standard data in machine learning methods are allowed to be incorrectly classified. On the other hand, the unreliable information (unlabeled data) in machine learning should be used in a semi-supervised way, i. e. they are taken as unlabeled samples which can provide useful information about sample distribution. In our model, soft constraints ensure that the useful prior information (directed and un-signed) is extracted while inaccurate information is filtered.

Next, we represent the hard constraints and soft constraints in matrix forms. Let the gold-standard directed and signed relationships be $K=(K_{ij})_{n\times n}$, which is an n×n matrix representing the known gene regulation information with signs. If the element K_{ij} is nonzero, it means that gene j has regulatory effect on gene i (activation or repression depends on the sign of K_{ij}, as determined by reliable biological experiments). The values for matrix K are set based on known information. Even though it is better to provide the quantitative strength of the known regulatory interactions in K, the vast majority of these are qualitative instead of quantitative in the databases or literature. In other words, one may know that gene i activates gene j, but the quantitative relationship is generally unavailable to depict how strong the activation is. In this case, we will decide final regulatory relationship from gene j to gene i from an LP-based algorithm by setting $K_{ij} >0$ or $K_{ij} <0$ as a hard constraint in the linear programming model. Here K serves as the gold standard positive data in the machine learning nomenclature, the difference is that we require the prior information in K to be correctly reflected in the final network structure.

There exists a second type of noisy prior regulatory information where the activation/repression role is unknown. We can represent such noisy information by soft constraints and store them into matrix $U=(U_{ij})_{n\times n}$, which is an n×n matrix representing the known gene regulation information without weights or signs. If the element U_{ij} is not zero, it means that gene j probably has regulatory effect on gene i (activation or repression is unknown and should be determined by data integration), and 0 if otherwise. We will incorporate U_{ij} into an LP-based algorithm as a soft constraint in our linear programming model by making all gene pairs for which U_{ij} is not zero free of regularization in the optimization process. If small molecule-protein interaction data are available, they can be incorporated by extending matrix U to n×(n+s) in a similar way.

In addition, we will treat the non-regulation relationship separately and store them into matrix $E=(E_{ij})_{n\times n}$, which is an n×n matrix and represents the known gene non-regulation information. If the element E_{ij} is zero, it means that gene i does not regulate gene j. Here E serves the similar role as the gold standard negative data in the machine learning meaning, the difference is prior information in E must be reflected in the final network structure. Because “gold standard” non-regulation relationships from biological experiments are often not published, negative examples need to be chosen with care. One possible selection method is to pick out the non-co-expression relationships from comprehensive expression compendium. The underlying assumption is that high quality non-regulatory relationships can be generated by considering pairs of genes whose expressions correlate weakly across various conditions. This can be further improved by combining several non-co-expression relationship detection methods together or using strict cutoffs. We will incorporate E_{ij} by using $E_{ij}=0$ as a hard constraint in our linear programming model.

In the following, we will discuss how to incorporate existing prior information into the inference of whole network by the LP-based algorithm.

Linear Programming Model for Data Integration

Assume that there are multiple microarray datasets for one organism, each of which corresponds to its own general solution in Equation (7). The next step is to find a consistent and also biologically plausible solution by determining variables Y^k, k=1,…, N. In (Y. Wang, Joshi, Zhang et al., 2006), we developed a method by exploiting L_1 norm in the formulation of the objective function to infer a sparse and consistent gene network. In this chapter, in addition to small molecule perturbations, we further consider the directed and signed regulatory relationship information K, directed and unsigned regulatory information U, and non-regulation relationships E. These new types of prior information are expected to improve the reliability of the inferred network and reduce the computational complexity.

Specifically, according to Equation (7), N networks can be separately inferred from N time-course datasets:

$$[J^k, P^k] = \frac{dX^k}{dt} U^k S^{k-1} V^{kT} = [\overline{J^k}, \overline{P^k}] + Y^k V^{kT} \tag{8}$$

where the superscript k=1,...,N is the index of the k-th dataset. Next, we will derive a sparse network structure $L=[J, P]=(L_{ij})_{n\times(n+s)}$ that is most consistent with $L^k=[J^k, P^k]= (L^k_{ij})_{n\times(n+s)}$ for k=1,...,N, as well as consistent with the directed and signed, directed and unsigned regulations, and non-regulatory relationships between genes. Mathematically the problem can be formulated as:

$$\begin{aligned} \min_{Y^1,Y^2,\ldots,Y^N,L} \quad & \sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{n+s} \omega_k \mid L_{ij} - L^k_{ij} \mid + \lambda \sum_{(i,j)\in\{(i,j)\mid K_{ij}=0 \text{ or } U_{ij}=0\}} \mid L_{ij} \mid \\ s.t. \quad & L_{ij} > 0 \quad \text{if} \quad K_{ij} > 0 \qquad i,j \in \{1,2,\ldots,n\} \\ & L_{ij} < 0 \quad \text{if} \quad K_{ij} < 0 \qquad i,j \in \{1,2,\ldots,n\} \\ & L_{ij} = 0 \quad \text{if} \quad E_{ij} = 0 \qquad i,j \in \{1,2,\ldots,n\} \end{aligned} \tag{9}$$

where L^k_{ij} is a function of Y^k, and $Y=(Y^1,...,Y^N)$. The objective function has two terms. The first term is a matching term which forces the matching of L and L^k, whereas the second term is a sparseness term which forces L to be sparse as a result of the minimization of the sum of L_1 norm.λis a positive parameter, which balances the matching and sparseness terms in the objective function. Here the soft constraints are added into the objective function in an implicit way, by removing the related sparseness terms of the objective function in Equation (9). The hard constraints are added as inequality or equality constraints in an explicit way. The first and second constraints are used to add the directed and signed information, and the third one is used to incorporate the non-regulatory relationship information.

The variables in (9) are L_{ij} and all of nonzero Y^k_{ij}. ω^k is a positive weight coefficient for the k-th dataset and $\sum^N_{k=1}\omega^k$ =1. Since different datasets may have different data qualities (e.g., different technologies, the number of repeats in measurements, etc.), the weight coefficient is used to represent the reliability of each dataset. The optimization problem (9) is an LP with L_1 norm, which is a well-studied problem. It is known that L_1 gives a more robust answer compared with L_2. The L_1-norm is more robust to outliers than the L_2-norm and does not penalize large deviations as much as the L_2-norm. As a result, the L_1-norm pays less attention to the parts of the regulatory interactions that are very different, and focuses more on the parts of the regulatory interactions that are conserved. As a result, this measure

is less sensitive toward noise and more robust towards outliers. Generally the optimal solution of (9) sets as many $|L_{ij}- L^k_{ij}|$ and $|L_{ij}|$ to zero as possible, thus ensuring a consistent and sparse structure for the inferred gene regulatory network.

As discussed previously, most documented regulation information is qualitative rather than quantitative. Therefore, we add the first and second inequality constraints of Equation (9) as hard constraints according to its activation or repression role stored in matrix K, and the strength of regulation is decided from the optimization algorithm. For example, add $L_{ij} <0$ if a repression relationship is known as $K_{ij} <0$ and derive the value of L_{ij} from the optimization process. In addition, the corresponding gene pair is removed from the second term (regularization term) in the objective function. We can also add the equality constraints $L_{ij}=0$ to Equation (9) to take into account the non-regulatory data. In addition, we also encode prior information in U as soft constraints in the following way. Specifically, for a gene pair where U_{ij} is non-zero (meaning that there probably exists regulatory relationship between the gene pair), we implement a soft constraint by removing the corresponding element of L_{ij} from the second term of objective function so that it is not subject to regularization in the optimization process. In this way, these regulatory interactions may be present in the optimal solution with signs and weights learned from the optimization process. In cases where the optimization process assigns zero weight to a gene pair, we assume that the prior information is probably noisy and is therefore ignored by the algorithm. The final result depends on the consistency of this information with microarray datasets or other prior information. It is reasonable since this prior information may or may not be correct, and therefore should be further filtered.

In Equation (9), each one of the matrices L, Y^1, Y^2, …, Y^N has almost n^2 variables. Thus the total number of variables is about n^3. For a gene regulatory network with 100 genes, even without prior information and other variables such as drug targets, the LP problem has 1,000,000 variables. To solve Equation (9) efficiently, a decomposition algorithm is used based on the special structure of Equation (9). This is done by iteratively solving the following two subproblems. We first fix L to solve N small-sized matching subproblems LP^1, LP^2, …, LP^N, followed by updating L by solving Equation (9) with fixed Y^1, Y^2, …, Y^N from the N subproblems. The procedure is repeated until convergence. The two decomposed subproblems are described in detail as follows.

- Subproblem-1: Set $L^k(q)=L^k(q-1)+ Y^k(q)V^{kT}$. At iteration q, obtain $Y^k_{ij}(q)$ by solving subproblems LP^1, LP^2, …, LP^N,

$$\min_{Y^1,Y^2,\ldots,Y^N} \quad \sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{n+s}\omega_k \mid L_{ij}(q-1) - L^k_{ij}(q) \mid + \lambda \sum_{(i,j)\in\{(i,j)|K_{ij}=0 \text{ or } U_{ij}=0\}} \mid L_{ij}(q-1) \mid \qquad (10)$$

where $L_{ij}(q-1)$ is fixed.

- Subproblem-2: At iteration q, obtain $L_{ij}(q)$ by solving the following LP with all of $Y^k_{ij}(q)$ and $L^k(q)$ fixed from Subproblem-1,

$$\begin{aligned}
\min_{L} \quad & \sum_{k=1}^{N}\sum_{i=1}^{n}\sum_{j=1}^{n+s}\omega_k \mid L_{ij}(q) - L_{ij}^{k}(q) \mid + \lambda \sum_{(i,j)\in\{(i,j)\mid K_{ij}=0 \text{ or } U_{ij}=0\}} \mid L_{ij}(q) \mid \\
s.t. \quad & L_{ij}(q) > 0 \quad \text{if} \quad K_{ij} > 0 \qquad i,j \in \{1,2,...,n\} \\
& L_{ij}(q) < 0 \quad \text{if} \quad K_{ij} < 0 \qquad i,j \in \{1,2,...,n\} \\
& L_{ij}(q) = 0 \quad \text{if} \quad E_{ij} = 0 \qquad i,j \in \{1,2,...,n\}
\end{aligned} \tag{11}$$

Although the solution depends on λ, λ is the only parameter that needs to be tuned. The procedures of solving Equations (10) and (11) and the choice of parameter λ are similar to (Y. Wang, Joshi, Zhang et al., 2006).

Compared with un-constrained LP model in (Y. Wang, Joshi, Zhang et al., 2006), the constraints in the above LP provide a consistent way to integrate all kinds of prior information. Specifically we incorporate reliable signed, noisy unsigned, and non-regulatory data in a systematic way. Given the nature of the gene regulatory network inference problem is under-determined (In other words, the number of variables far exceeds the equations for which variables are related), the proper incorporation of the prior information from heterogeneous data sources improves the reconstruction accuracy.

It should be noted that the above methodology has three advantages in terms of both model and algorithm. Firstly, the variables L_{ij} in Equation (9) include not only the connectivity matrix of genes which represents the effect of activation, no interaction and repression, but also the connectivity matrix of perturbations which represents the effect of the small-molecule perturbations on genes. This is very important because our method is able to properly identify the target genes of perturbations and thus has the potential to be applied to the drug design and mechanism of action discovery of molecules. Secondly, the objective function has both sparse and non-sparse terms. The non-sparse term is used to represent the interactions or effects among genes or between external inputs and genes based on the noisy prior information or experimental data. In this way, the soft constraints are considered and added in a consistent manner. Thirdly, the new model can improve reconstruction accuracy by introducing hard constraints on "gold-standard" prior information.

From the algorithmic and computational efficiency aspect, Equation (9) is a constrained L_1 linear approximate problem, in contrast to the linear regression model of (Y. Wang, Joshi, Zhang et al., 2006). For the first subproblem, an efficient primal algorithm can be designed by taking advantage of the special structure of the linear programming formulation of the L_1 problem; for the second subproblem, it can be decomposed as a series of constrained and unconstrained small-scale linear programming (Y. Wang, Joshi, Zhang et al., 2006) and the problem can be solved efficiently.

The data integration strategy in this chapter is different from the supervised inference methods (T. Kato, Tsuda, & Asai, 2005) which adopt the kernel matrix representation of networks and integrate different biological data in a simple weighted sum. In this chapter the gene regulations are derived from time-course data by differential equations instead of similarity evaluation in kernel matrix. Specifically, the "gold standard" prior information is expressed as hard constraints or soft constraints (i.e., sparse term in L_{ij} of the objective function) in the LP formulation, depending on the certainty or reliability of the information. Thus, we can obtain the most consistent solution among multiple datasets by satisfying those constraints. In particular, in our prior information learning framework, our method ensures all of the hard constraints to hold, and prefers the soft constraints to hold, but the regulatory interactions corresponding to soft constraints may or may not hold depending on their consistency with other data. Therefore, if prior information is not reflected in the optimal solution, it is because this prior informa-

tion is inconsistent with the microarray datasets and other information. As such, the proposed algorithm can also filter out the noise in prior information based on the requirement of consistency among all data sources.

RESULTS

In this section, we first report a simulated numerical example to validate our method. Then we apply our method to a real experimental data to reconstruct yeast gene regulatory network. We show that our method is effective in recovering the network connectivity from integrated data sources. Importantly, with supervised information, our method can infer the network structure and further identify the compound targets in a more accurate and reliable manner.

Simulated Example

The first example is a small simulated network to demonstrate the usefulness of data integration and prior information in the network inference method. We constructed a small regulatory network with six genes governed by:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \\ \dot{x}_4(t) \\ \dot{x}_5(t) \\ \dot{x}_6(t) \end{bmatrix} = \begin{bmatrix} -1.0 & 0.0 & 0.01 & 0.0 & 0.03 & 0.03 \\ 0.2 & -1.2 & 0.0 & 0.4 & -0.05 & 0.0 \\ 0.0 & 0.0 & -1.0 & 0.0 & 0.0 & -0.05 \\ 0.0 & -0.05 & 0.0 & -1.5 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.0 & -1.2 & 0.0 \\ 0.0 & 0.03 & 0.0 & -0.01 & 0.0 & -1.0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix} + \begin{bmatrix} 2.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{bmatrix} u_0 \qquad (12)$$

where x_i reflects the expression level of the gene-i for i=1,...,6. One perturbation (s=1) is applied to the first gene, which is indicated by $P=[2.0,0.0,0.0,0.0,0.0,0.0]^T$. P has all its elements equal to 0 except the element for the gene that is the direct target of the perturbation. u_0 contains the detailed information about the perturbation, which can be either time-independent or time-dependent.

We generate four time-course datasets in different conditions. Every dataset has five time points and the time points are equally-spaced from the start to the end. These datasets differ in the choice of perturbation and time step. The first dataset is obtained by taking perturbation u_0=1 as a constant and the time step is 0.1. The second dataset is also obtained by taking u_0=1 as a constant but the time step is 0.15. For the third dataset, perturbation varies with time and gradually increase from u_0=1 to u_0=2, and the time step is 0.2. The fourth dataset is obtained without perturbation and with time step 0.2. The initial values of the system are randomly generated from [1.0, 1.1] and the Gaussian noise is added to the data matrix with zero mean and fixed standard deviation $\sigma=0.2\|X\|$, where $\|X\|$ is the L_∞ norm of the data matrix X. In the following, we will show that the datasets can be combined together to infer the gene regulatory network by our method. The parameter λ is set to 0.1 to make the inferred network sparse. The supervised information K is denoted by the following matrix,

$$K = \begin{bmatrix} -1.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.03 \\ 0.0 & 0.0 & 0.0 & 0.4 & -0.05 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.0 \\ 0.0 & -0.05 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (13)$$

where nonzero elements will be added in the LP as constraints.

This simulated example illustrates the three advantages of our method for reconstructing gene regulatory networks. Firstly, our method can identify more correct regulatory relationships among genes. The numerical results are depicted in Figure 2, which shows the true network and the reconstructed networks without and with supervised information, respectively. In the case without supervised information, 4 edges are identified correctly out of 7 predicted nonzero edges. In contrast, in the case with supervised information, 11 edges are identified correctly out of 16 predicted nonzero edges. Thus the prediction accuracy is improved from 57.14% to 68.75%.

Secondly, the inferred network by our method is quantitatively more accurate. We use the following indices E_1 and E_2 to assess the prediction accuracy:

$$E_1 := \sum_{i=1}^{n} \sum_{j=1}^{n} \left| J_{ij}^T - J_{ij}^R \right| \quad (14)$$

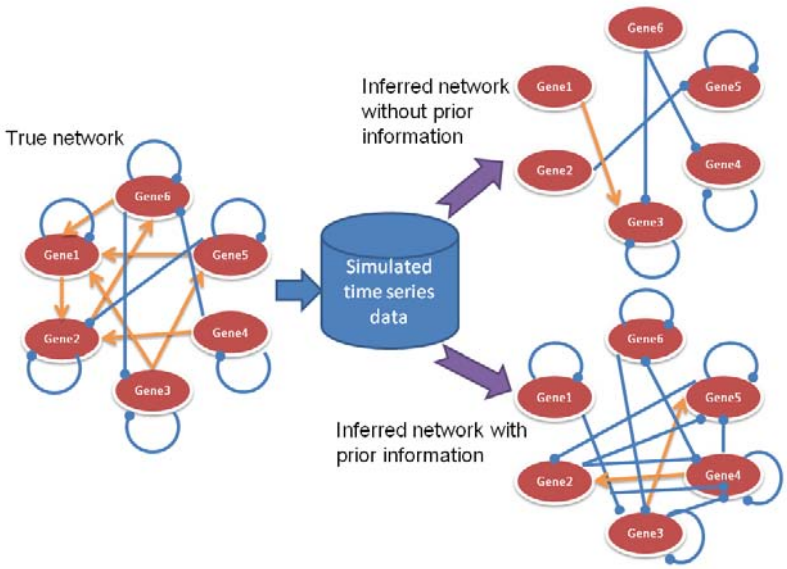
$$E_2 := \sum_{i=1}^{n} \sum_{j=1}^{n} \left(J_{ij}^T - J_{ij}^R \right)^2 \quad (15)$$

where J_{ij}^T and J_{ij}^R are interaction strength from gene-j to gene-i for the true and inferred networks, respectively. We found that adding supervised information reduces the inference error. For example, E_1 decreases by 0.7139 (excluding the error reduction 1.73 due to the knowledge of K) and E_2 decreases by 1.4857 (excluding the error reduction 1.20 due to the knowledge of K).

Thirdly, our method makes more accurate predictions about the targets genes of perturbation. This is very important as our method has the potential to be applied to the drug design and function discovery of molecules. According to the computational results, the inferred perturbation vector is P = [0.35, -0.08, 0.09, 0.0, 0.0, 0.0] without supervised information, whereas the prediction results are improved to P = [1.25, 0.0, -0.01, 0.0, 0.0, -0.03] with supervised information. These results show that our method can correctly identify the first gene to be the direct target of the applied perturbation, and the knowledge of the supervised information can helps reduce inference error.

There are two reasons for the accurate inference by our method. The first reason is the contribution of multiple datasets. By combining the time-course datasets of different types and in different perturbation conditions, more information are utilized and the problem of high dimensionality is significantly alleviated (refer to (Y. Wang, Joshi, Zhang et al., 2006) for details). The second reason is the contribution of prior information. Due to the scarcity of gene expression data and the high-dimensionality of the gene network parameter space, the problem of gene network inference is fundamentally under-determined. The supervised information help reduce the intrinsic dimensionality of the search space dramatically, thus making the inferred network more accurate both qualitatively and quantitatively.

Figure 2. Regulatory network reconstruction for the simulated example with 6 genes. Red arrows represent activation, and blue arcs represent repression.



Combining ChIP-Chip and Expression Data to Infer Gene Regulatory Network in Yeast

We combined gene expression data with ChIP-chip data to infer gene regulatory network in yeast. As mentioned above, the ChIP-chip methodology involves the chromatin immunoprecipitation of an epitope-tagged TF bound to DNA fragments containing target promoters, followed by the hybridization of those amplified DNA fragments to an intergenic microarray (Lee et al., 2002). ChIP-chip data have the advantage that they provide a direct biochemical link between TFs and promoters and have the potential to identify targets without knowing the activating conditions. From this viewpoint, ChIP-chip data are an important source of information for direct transcriptional regulatory interactions. In this example we show the network reconstruction accuracy can be improved by incorporating TF DNA binding data (ChIP-chip data) into our model as prior information (soft constraints). We tested our method using the public time-series microarray data for cell cycle studies in *Saccharomyces cerevisiae* which are obtained from the Stanford Microarray Database (Demeter et al., 2007). We collected 4 datasets with different conditions (Response to Elutriation, 14 time points; Response to CDC15, 24 time points; Response to alpha factor fkh1, fkh2, 13 time points; Response to fkh1, fkh2,13 time points). Among all the yeast genes, 145 of them have changes of 2 fold up or down in at least 20% of the expression level across all datasets.

We added the TF-DNA binding data as prior information to infer the gene regulatory network in a more reliable manner. In (Lee et al., 2002), a genome-wide location analysis experiment was performed for 106 yeast TFs. From their supporting website (http://jura.wi.mit.edu/young_public/regulatory_network/), we downloaded the TF-target gene interactions and TF-TF interactions as prior information. As a summary, there are 75 TFs (in the list of 106 TFs of (Lee et al., 2002)) in the 145 gene list, and there

are a total of 161 known interactions including 93 known TF-TF interactions and 68 TF-target gene interactions. Furthermore, we manually selected 22 interactions with known activating or repressing conditions by checking the GO database.

Then we apply our method to these real experimental data in yeast. There are two kinds of prior information. One is the 22 interactions with known activating or repressing conditions which can be directly added as the hard constraints in the LP model. The remaining 139 known interactions without activating or repressing information are taken as soft constraints for which we simply remove their corresponding sparse terms from the objective function of the LP model.

When $\lambda=0$, we obtained 622 interactions. All the 161 known interactions are correctly inferred. Among them, 22 interactions with known activating or repressing conditions are correctly inferred and the activating or repressing conditions of the remaining 139 interactions are predicted by our method. Among the newly inferred 461 edges, validation results by YEASTRACT database (Teixeira et al., 2006) show that there are 11 documented interactions and there are 66 edges identified as potential interactions, for which transcription factors have at least 1 binding site in the promoter regions of their target genes.

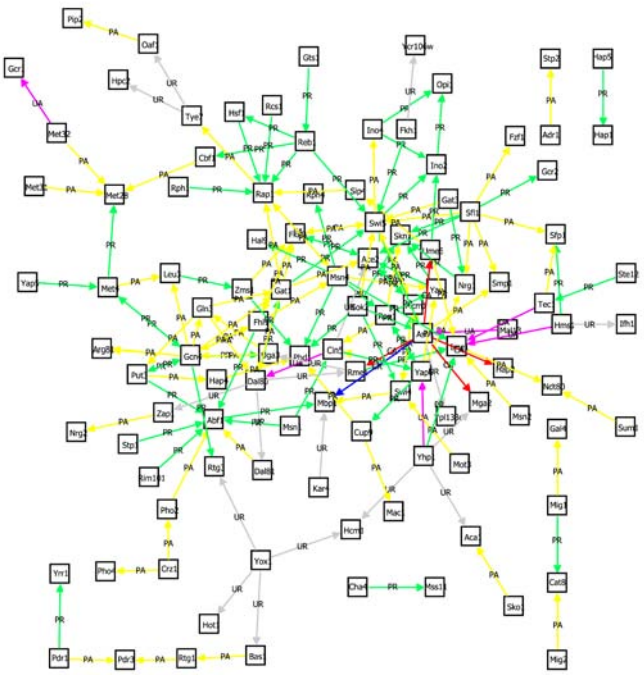
When we set the parameter $\lambda=0.1$ to make the inferred network sparse, a total of 219 interactions were inferred with predicted activating or repressing conditions. Again all the 161 known interactions (Among them, 22 interactions have known activating or repressing conditions) are correctly inferred. The validation results by YEASTRACT show that in the newly inferred 58 edges, 5 are documented and 11 are identified as potential interactions. In Figure 3, we draw the reconstructed gene regulatory network when $\lambda=0.1$ (self regulatory interactions are not shown).

In this experiment, we combine the time-series microarray data and genomic location data to infer whether a regulator acts as an activator or repressor. Generally, we can use genomic location data to infer the presence of regulators at promoters, but we cannot determine the type of TF-target gene interactions. By further combining gene expression data, our method can infer not only the existence of regulatory interactions between TFs and target genes, but also the sign of the regulation (positive or negative). From this example, we can also see that computational method is complementary to experimental methods, e.g., it provides information whether a TF is an activator or repressor by its regulatory role based on the dynamic behavior of the gene expression.

CONCLUSION

Our proposed data integration and network reconstruction method in this chapter not only improves the reliability of the inferred gene regulatory network, but also can be applied to drug design and many other areas of biomedical research and bioengineering. Specifically, we propose to combine computational analysis of multiple microarray datasets and other types of biological experiments together for inferring gene regulatory network and further identifying small molecule targets of perturbation experiments. The proposed algorithm is mainly based on linear programming framework with the variables representing the regulatory relationships among genes and small molecule-protein interactions. Available information from heterogeneous data sources is incorporated into the LP as constraints. They can be divided into two classes according to data reliability. For example, the regulatory relationships mined from literature and databases are more reliable and we know exactly the regulator gene, target gene, and the activation or repression role. Hence, they can be treated as hard constraints in our linear programming which will be strictly reinforced. On the other hand, the co-expression data, ChIP-chip data and protein-protein

Figure 3. Inferred gene regulatory network when λ=0.1 by combining expression data with TF-DNA binding data. Self regulatory interactions are not shown. Known activations are shown in yellow with label 'PA'. Known repressions are shown in green with label 'PR'. Newly inferred activations that are subsequently confirmed are shown in red with label 'CA'. Newly inferred repressions that are subsequently confirmed are shown in blue with label 'CR'. Newly inferred activations and repressions that are yet to be confirmed are shown in pink with label 'UA' and gray with label 'UR', respectively.



interaction data are generally noisy and the related pairs often are unsigned, meaning that the activation or repression role is unknown. In this case, we treat them as soft constraints, which may or may not be satisfied. In this way, our linear programming model provides a flexible prior information learning framework. It finds the most consistent gene regulatory network by balancing among heterogeneous data sources. One major advantage of the proposed method is that it theoretically ensures the derivation of the most consistent network with respect to the available datasets or information, thereby alleviating the problem of data scarcity and improving the reliability. In addition, this algorithm allows us to infer small molecule targets by integrating perturbation experiments, and holds the promise for applications in drug design and other areas in biomedical engineering.

FUTURE RESEARCH DIRECTIONS

With rapid advances of various high-throughput experimental techniques, more and more biological data are increasingly available. Thus it is now possible to quantitatively study regulation interactions in a systematic way. Generally speaking, there are three kinds of regulatory relationships among the

regulators (transcriptional factors and cofactors) and target genes. They are the relationships between target genes, the relationships between regulators, and the relationships between regulators and target genes. Network reconstruction aims to reveal regulatory mechanisms by inferring these relationships from biological data.

The mapping of the gene regulatory network—the set of interactions among all genes in the genome—is one of the most difficult tasks in molecular biology. For example, there are 6000 genes in yeast, and as a result there are at least 18 millions parameters to be determined in our linear differential model. In contrast, the mapping of the transcriptional regulatory network — the set of all physical interactions among transcriptional factors and their target genes — has much less parameters from the computational viewpoint. There are about 200 transcriptional factors in yeast and 6000 target genes and thus there are about 1.2 million parameters to be determined. Compared to the above two tasks, the reconstruction of the transcriptional factor interaction network is perhaps the easiest. Since there are about 200 transcriptional factors in yeast, we only need to determine about 20,000 parameters. It is well known that the transcription factor sub-proteome is very important for gene regulation and especially difficult for experimental characterization. Hence, the computational methodology to predict these regulatory subnetworks among TFs is crucial. In the case of the transcription factor interactome, transcriptional regulation in eukaryotes occurs through the coordinated action of multiple transcription factors. So combinatorial regulation is a primary mechanism for achieving fine-tuned transcriptional control, is an important component of the mechanisms of action for many biologically active small molecules, and holds the promise to reveal the complexity of gene regulation mechanisms (Balaji, Babu, Iyer, Luscombe, & Aravind, 2006; Bluthgen, Kielbasa, & Herzel, 2005; Chang, Wang, & Chen, 2006; M. Kato, Hata, Banerjee, Futcher, & Zhang, 2006).

The linear differential equation model in this chapter makes the important assumption that the structure of the regulatory network is stationary, and does not `rewire' under the environmental conditions for those different datasets. This means that the change of environmental conditions is assumed to alter the level of gene expression instead of the network structure. Obviously this is not true in reality. One of the future research directions is to reverse engineer the network architecture from time-series microarray data based on a nonlinear differential equation model, which will capture the complex and nonlinear properties in gene regulatory process but will involve more parameters.

Data integration is still a very challenging problem. A complex network reconstruction methodology needs high resolution datasets so as to accurately infer the network structure. Here high-resolution data mean high-quality time-course microarray data which are expected to capture the dynamic behavior of the gene regulatory networks and also the conditional responsive transcription factor-DNA and protein-protein interaction data. As a result, sophisticated data integration techniques play a key role.

Recently Faith et al. assembled 445 Escherichia coli microarrays to address this issue and demonstrated an unsupervised network inference method, called context likelihood of relatedness (CLR), which uses transcriptional profiles of an organism across a diverse set of conditions to systematically determine transcriptional regulatory interactions (Faith et al., 2007). By generating a compendium of microarrays, they showed that it is possible to infer a high-precision regulatory map and simultaneously obtain rich data on condition-specific regulation. The strategy here is simple: it assembles all the microarray datasets to a profile or compendium and applies algorithms based on correlation coefficients or mutual information measure, such as Relevance network (Butte et al., 2000), ARACNe (Margolin et al., 2006), and CLR (Faith et al., 2007) to find the co-expression or co-regulation relationships. This strategy can be potentially enhanced in several ways. First, the existing methods treat a set of time-course data

points independently and ignore the dynamic property of gene regulation process. Second, the existing methods can only identify whether two genes have regulatory relationships, but cannot provide the detailed information about regulatory roles such as activation or repression. The existing methods can be improved by considering as much dynamic information as possibly when integrating time-course microarray datasets.

We believe that data integration and network reconstruction should be conducted in a simultaneous way. We should determine the data integration parameters and the network structure parameters together. In this way, the solution will be expected to be globally optimal and the most consistent. In this chapter we provided such a model. It can be further extended to a more general model to assign weights to different sources of data. In the future, we will extend the current work of revealing the complex mechanisms of transcriptional control in two ways. First, regulatory network reconstruction can be greatly improved by the integration of more diverse genomic datasets such as sequence, protein structure, gene expression, TF-DNA interaction, non-coding RNA-mRNA interaction, protein-protein interaction, and metabolic reaction data. Second, transcriptional regulatory processes can be more accurately modeled by taking into account cooperativity among individual proteins, nonlinearity, and dynamic behaviors.

ACKNOWLEDGMENT

YW, RSW, XSZ, and LC are supported by JSPS and NSFC under JSPS-NSFC collaboration project. YW is also supported by National Natural Science Foundation of China under Grant No.10701080 and No. 10801131.YX is supported by a Research Starter Grant in Informatics from the PhRMA Foundation.

REFERENCES

Alaoui-Ismaili, M. H., Lomedico, P. T., & Jindal, S. (2002). Chemical genomics: discovery of disease genes and drugs. *Drug Discovery Today, 7*(5), 292–294. doi:10.1016/S1359-6446(02)02185-2

Balaji, S., Babu, M. M., Iyer, L. M., Luscombe, N. M., & Aravind, L. (2006). Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast. *Journal of Molecular Biology, 360*(1), 213–227. doi:10.1016/j.jmb.2006.04.029

Benson, M. L., Smith, R. D., Khazanov, N. A., Dimcheff, B., Beaver, J., & Dresslar, P. (2007). Binding MOAD, a high-quality protein ligand database. *Nucleic Acids Research, 36*(Database issue), D674–D678. doi:10.1093/nar/gkm911

Bluthgen, N., Kielbasa, S. M., & Herzel, H. (2005). Inferring combinatorial regulation of transcription in silico. *Nucleic Acids Research, 33*(1), 272–279. doi:10.1093/nar/gki167

Brazma, A., Jonassen, I., Vilo, J., & Ukkonen, E. (1998). Predicting gene regulatory elements in silico on a genomic scale. *Genome Research, 8*(11), 1202.

Breitkreutz, B. J., Stark, C., Reguly, T., Boucher, L., Breitkreutz, A., & Livstone, M. (2008). The BioGRID interaction database: 2008 update. *Nucleic Acids Research, 36*(Database issue), D637. doi:10.1093/nar/gkm1001

Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R., & Kohane, I. S. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(22), 12182–12186. doi:10.1073/pnas.220392197

Chang, Y.-H., Wang, Y.-C., & Chen, B.-S. (2006). Identification of transcription factor cooperativity via stochastic system model. *Bioinformatics (Oxford, England)*, *22*(18), 2276–2282. doi:10.1093/bioinformatics/btl380

D'Haeseleer, P., Liang, S., & Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics (Oxford, England)*, *16*(8), 707–726. doi:10.1093/bioinformatics/16.8.707

De Hoon, M. J. L., Imoto, S., Kobayashi, K., Ogasawara, N., & Miyano, S. (2003). Inferring gene regulatory network from time-ordered gene expression data of bacillus subtilis using differential equations. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 17–28.

Demeter, J., Beauheim, C., Gollub, J., Hernandez-Boussard, T., Jin, H., & Maier, D. (2007). The Stanford Microarray Database: implementation of new analysis tools and open source release of software. *Nucleic Acids Research*, *35*(Suppl 1), D766–D770. doi:10.1093/nar/gkl1019

Dewey, T. G., & Galas, D. J. (2001). Dynamic models of gene expression and classification. *Functional & Integrative Genomics*, *1*(4), 269–278. doi:10.1007/s101420000035

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., & Cottarel, G. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology*, *5*(1), e8. doi:10.1371/journal.pbio.0050008

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*(5659), 799–805. doi:10.1126/science.1094068

Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *301*(5629), 102–105. doi:10.1126/science.1081900

Gardner, T. S., & Faith, J. J. (2005). Reverse-engineering transcription control networks. *Physics of Life Reviews*, *2*(1), 65–88. doi:10.1016/j.plrev.2005.01.001

Gustafsson, M., Hornquist, M., & Lombardi, A. (2005). Constructing and analyzing a large-scale gene-to-gene regulatory network-Lasso-Constrained inference and biological validation. [TCBB]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*(3), 254–261. doi:10.1109/TCBB.2005.35

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., & Danford, T. W. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, *431*, 99–104. doi:10.1038/nature02800

Holter, N. S., Maritan, A., Cieplak, M., Fedoroff, N. V., & Banavar, J. R. (2001). Dynamic modeling of gene expression data. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(4), 1693. doi:10.1073/pnas.98.4.1693

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics (Oxford, England)*, *19*(17), 2271–2282. doi:10.1093/bioinformatics/btg313

Hwang, D., Rust, A. G., Ramsey, S., Smith, J. J., Leslie, D. M., & Weston, A. D. (2005). A data integration methodology for systems biology. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(48), 17296–17301. doi:10.1073/pnas.0508647102

Joyce, A. R., & Palsson, B. O. (2006). The model organism as a system: integrating Omics data sets. *Nature Reviews. Molecular Cell Biology*, *7*(3), 198–210. doi:10.1038/nrm1857

Kato, M., Hata, N., Banerjee, N., Futcher, B., & Zhang, M. Q. (2006). Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biology*, *5*, R56. doi:10.1186/gb-2004-5-8-r56

Kato, T., Tsuda, K., & Asai, K. (2005). Selective integration of multiple biological data for supervised network inference. *Bioinformatics (Oxford, England)*, *21*(10), 2488–2495. doi:10.1093/bioinformatics/bti339

Kuhn, M., von Mering, C., Campillos, M., Jensen, L. J., & Bork, P. (2008). STITCH: Interaction networks of chemicals and proteins. *Nucleic Acids Research*, *36*(Database issue), D684. doi:10.1093/nar/gkm795

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., & Gerber, G. K. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science*, *298*(5594), 799–804. doi:10.1126/science.1075090

Liu, T., Lin, Y., Wen, X., Jorissen, R. N., & Gilson, M. K. (2007). BindingDB: A Web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research*, *35*(Suppl 1), D198–D201. doi:10.1093/nar/gkl999

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., & Dalla Favera, R. (2006). ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, *7*(Suppl 1), S7. doi:10.1186/1471-2105-7-S1-S7

Nachman, I., Regev, A., & Friedman, N. (2004). Inferring quantitative models of regulatory networks from expression data. *Bioinformatics (Oxford, England)*, *20*(Suppl 1), i248–i256. doi:10.1093/bioinformatics/bth941

Nariai, N., Tamada, Y., Imoto, S., & Miyano, S. (2005). Estimating gene regulatory networks and protein-protein interactions of Saccharomyces cerevisiae from multiple genome-wide data. *Bioinformatics (Oxford, England)*, *21*(Suppl 2), ii206–ii212. doi:10.1093/bioinformatics/bti1133

Tegner, J., Yeung, M. K., Hasty, J., & Collins, J. J. (2003). Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(10), 5944. doi:10.1073/pnas.0933416100

Teixeira, M. C., Monteiro, P., Jain, P., Tenreiro, S., Fernandes, A. R., & Mira, N. P. (2006). The YEASTRACT database: A tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. *Nucleic Acids Research*, *34*(Suppl 1), D446–D451. doi:10.1093/nar/gkj013

Wang, R. S., Wang, Y., Zhang, X. S., & Chen, L. (2007). Inferring transcriptional regulatory networks from high-throughput data. *Bioinformatics (Oxford, England)*, *23*(22), 3056–3064. doi:10.1093/bioinformatics/btm465

Wang, Y., Joshi, T., Xu, D., Zhang, X., & Chen, L. (2006). *Supervised inference of gene regulatory networks by linear programming*. (. *LNCS*, *4115*, 551.

Wang, Y., Joshi, T., Zhang, X. S., Xu, D., & Chen, L. (2006). Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics (Oxford, England)*, *22*(19), 2413. doi:10.1093/bioinformatics/btl396

Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., & Tzur, D. (2007). DrugBank: A knowledgebase for drugs, drug actions, and drug targets. *Nucleic Acids Research*, *36*, D901–D906. doi:10.1093/nar/gkm958

Yeung, M. K., Tegner, J., & Collins, J. J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(9), 6163. doi:10.1073/pnas.092576199

ADDITIONAL READING

Alon, U. (2007). *An introduction to systems biology: Design principles of biological circuits*. Chapman & Hall/CRC.

Alvarez-Buylla, E. R., Beniez, M., & Daila, E. B., Chaos, Espinosa-Soto, C., & Padilla-Longoria, P. (2007). Gene regulatory network models for plant development. *Current Opinion in Plant Biology*, *10*(1), 83–91. doi:10.1016/j.pbi.2006.11.008

Bansal, M., Gatta, G. D., & di Bernardo, D. (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics (Oxford, England)*, *22*(7), 815–822. doi:10.1093/bioinformatics/btl003

Bonneau, R., Facciotti, M. T., Reiss, D. J., Schmid, A. K., Pan, M., & Kaur, A. (2007). A predictive model for transcriptional control of physiology in a free living cell. *Cell*, *131*(7), 1354–1365. doi:10.1016/j.cell.2007.10.053

Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., & Baliga, N. S. (2006). The inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*, R36. doi:10.1186/gb-2006-7-5-r36

Bornholdt, S. (2005). Systems biology: Less is more in modeling large genetic networks. *Science*, *310*(5747), 449–451. doi:10.1126/science.1119959

di Bernardo, D., Thompson, M. J., Gardner, T. S., Chobot, S. E., Eastwood, E. L., & Wojtovich, A. P. (2005). Chemogenomic profiling on a genomewide scale using reverse-engineered gene networks. *Nature Biotechnology*, *23*, 377–383. doi:10.1038/nbt1075

Driscoll, M. E., & Gardner, T. S. (2006). Identification and control of gene networks in living organisms via supervised and unsupervised learning. *Journal of Process Control*, *16*(3), 303–311. doi:10.1016/j.jprocont.2005.06.010

Ernst, J., Vainas, O., Harbison, C. T., Simon, I., & Bar-Joseph, Z. (2007). Reconstructing dynamic regulatory maps. *Molecular Systems Biology*, *3*(74), 1–13.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*(5659), 799–805. doi:10.1126/science.1094068

Gustafsson, M., Hornquist, M., & Lombardi, A. (2005). Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation. [TCBB]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*(3), 254–261. doi:10.1109/TCBB.2005.35

Hartemink, A. J. (2005). Reverse engineering gene regulatory networks. *Nature Biotechnology*, *23*, 554–555. doi:10.1038/nbt0505-554

Hayete, B., Gardner, T. S., & Collins, J. J. (2007). Size matters: Network inference tackles the genome scale. *Molecular Systems Biology*, *3*, 77. doi:10.1038/msb4100118

Markowetz, F., & Spang, R. (2007). Inferring cellular networks: A review. *BMC Bioinformatics*, *8*(Suppl 6), S5. doi:10.1186/1471-2105-8-S6-S5

Miyano, S. (2003). Use of gene networks for identifying and validating drug targets. *Journal of Bioinformatics and Computational Biology*, *1*(3), 459–474. doi:10.1142/S0219720003000290

Palsson, B. O. (2006). *Systems biology: Properties of reconstructed networks*. New York: Cambridge University Press.

Pan, Y., Durfee, T., Bockhorst, J., & Craven, M. (2007). Connecting quantitative regulatory-network models to the genome. *Bioinformatics (Oxford, England)*, *23*(13), i367. doi:10.1093/bioinformatics/btm228

Quach, M., Brunel, N., & d'Alche-Buc, F. (2007). Estimating parameters and hidden variables in non-linear state-space models based on ODEs for biological networks inference. *Bioinformatics (Oxford, England)*, *23*(23), 3209. doi:10.1093/bioinformatics/btm510

Reiss, D., Baliga, N., & Bonneau, R. (2006). Integrated biclustering of heterogeneous genomewide datasets for the inference of global regulatory networks. *BMC Bioinformatics*, *7*(1), 280. doi:10.1186/1471-2105-7-280

Rosenfeld, S. (2007). Stochastic cooperativity in non-linear dynamics of genetic regulatory networks. *Mathematical Biosciences*, *210*(1), 121–142. doi:10.1016/j.mbs.2007.05.006

Schlitt, T., & Brazma, A. (2007). Current approaches to gene regulatory network modelling. *BMC Bioinformatics*, *8*(Suppl 6), S9. doi:10.1186/1471-2105-8-S6-S9

Soranzo, N., Bianconi, G., & Altafini, C. (2007). Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: Synthetic vs. real data. *Bioinformatics (Oxford, England)*, *23*(13), 1640. doi:10.1093/bioinformatics/btm163

Sperling, S. (2007). Transcriptional regulation at a glance. *BMC Bioinformatics*, *8*(Suppl 6), S2. doi:10.1186/1471-2105-8-S6-S2

Steinke, F., Seeger, M., & Tsuda, K. (2007). Experimental design for efficient identification of gene regulatory networks using sparse Bayesian models. *BMC Systems Biology*, *1*(51).

Sun, L., Jiang, L., Li, M., & He, D. (2006). Statistical analysis of gene regulatory networks reconstructed from gene expression data of lung cancer. *Physica A: Statistical Mechanics and its Applications, 370*(2), 663-671.

Wang, Y., Zhang, X.-S., & Chen, L. (2009). A network biology study on circadian rhythm by integrating various omics data. *OMICS: A Journal of Integrative Biology*, *13*(4), 313–324. doi:10.1089/omi.2009.0040

Zhang, H., Pu, J., & Zhang, J. (2006). Construction of gene regulatory networks based on gene ontology and multivariable regression. *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation* (pp. 1324-1328).

Zhao, W., Serpedin, E., & Dougherty, E. R. (2006). Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics (Oxford, England)*, *22*(17), 2129. doi:10.1093/bioinformatics/btl364

Zou, M., & Conzen, S. D. (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics (Oxford, England)*, *21*(1), 71–79. doi:10.1093/bioinformatics/bth463

Chapter 20
Integrating Various Data Sources for Improved Quality in Reverse Engineering of Gene Regulatory Networks

Mika Gustafsson
Linköping University, Sweden

Michael Hörnquist
Linköping University, Sweden

ABSTRACT

In this chapter we outline a methodology to reverse engineer GRNs from various data sources within an **ODE** *framework. The methodology is generally applicable and is suitable to handle the broad error distribution present in microarrays. The main effort of this chapter is the exploration of a fully data driven approach to the* **integration** *problem in a "***soft evidence***" based way.* **Integration** *is here seen as the process of incorporation of uncertain a priori knowledge and is therefore only relied upon if it lowers the prediction error. An efficient implementation is carried out by a* **linear programming** *formulation. This LP problem is solved repeatedly with small modifications, from which we can benefit by restarting the primal* **simplex method** *from* **nearby solutions***, which enables a computational efficient execution. We perform a case study for data from the* **yeast cell cycle***, where all verified genes are putative regulators and the a priori knowledge consists of several types of binding data, text-mining and annotation knowledge.*

INTRODUCTION

Biological systems are intrinsically complex, still robust and at the same time able to quickly adapt to new situations. To understand, describe and model a wide range of biological systems −involving genes, proteins, metabolites and ecological food webs− networks have served as the unifying language (Barabasi *et al.* 2004). This description has often revealed a complex network topology. In the case of

DOI: 10.4018/978-1-60566-685-3.ch020

Gene Regulatory Networks (GRNs), some features are the existence of key genes regulating multiple processes ("hubs"), feed-back motifs and modularity enhancing the system robustness (Milo *et al.* 2002; Barabasi *et al.* 2004). Furthermore, the dynamical systems seem to be tuned to enable a stable system by keeping hubs repressed, but still flexible by utilizing, e.g., incoherent feed-back loops (Gustafsson *et al.* In press b, Ma'ayan *et al.* 2008). In addition to the architectural complications, we know that gene regulation is a non-linear process including combinatorial control, saturation and stochasticity. These pieces give raise to an extremely challenging modelling problem, which becomes even more complicated by the size of the genome.

Further, the experimental advancements in the last decades have resulted in a vast amount of large-scale data sets available through public databases. To infer a large-scale GRN it is of uttermost importance to take as much as possible of these data into account. Particularly informative for understanding genome-wide gene regulation is the interaction map between Transcription Factors (TFs) and their DNA binding regions. This information may give direct structural properties of the regulatory possibilities, e.g., the presence of a binding element upstream of gene of A for a TF which gene B codes for induces an enhanced possibility for regulation of gene A by gene B.

Other types of structural information may come from sequence based predictions, e.g., prediction of putative regulations from the TF binding sites (TFBS) and from common biological knowledge. The latter can be incorporated in a variety of ways, which may come from annotation knowledge or more "unclean" knowledge as text-mining. Annotation knowledge may be the collection of detailed knowledge from previous experiments, while text-mining may be a possibility to include the plethora of published biological papers in databases. On a more detailed causal level there is also a large number of time-series expression data sets for mRNA levels (see, e.g., Omnibus at Entrez (PubMed 2007) for collections at a unified format). However, although all these experiments are present on a large-scale, they are all typically several orders of magnitudes smaller than the number of presumptive regulators. Hence, all data at hand should be taken in consideration to overcome the indefiniteness of the reverse engineering problem. The greatest challenge in GRN inference to tackle is that the number of genes vastly exceeds the number of experiments, making it a tough statistical question. We should therefore strive to avoid introducing more entities in the model. Consequently, we project gene regulation onto the space of genes only, despite the fact that gene regulation is carried out from the interactions of mRNA molecules, proteins and metabolites (Brazhnik *et al.* 2002; Ptashne *et al.* 2002). Indeed, the obtained GRN is then an effective network of gene-to-gene interactions, where these interactions cannot be interpreted as biochemical reactions.

Reverse engineering of genome-scale GRNs is a grand challenge for system biologists, with a high potential for drug discovery. The challenge consists in taking many small pieces of information ranging from widely different experiment types and **prior knowledge** properly into account. However, most of the genome-wide experiments are associated with great uncertainties, thus connected with many false positive and negative regulations. Nevertheless, algorithms have gradually become more refined, from the first cluster analyses of gene expression data (Eisen *et al.* 1998), to more recent dynamic network inferences (Segal 2003; Luscombe *et al.* 2004; Bonneau *et al.* 2006; Gustafsson *et al.* 2005; Wang *et al.* 2006) taking more data into account (Luscombe *et al.* 2004; Bonneau *et al.* 2006). The next step to get more accurate descriptions of the GRNs is to carefully take different data sources into account, such as TF-bindings, protein-interactions, sequence information, literature knowledge and of course expression data. The introduction of several data types in the reverse engineering process enforces a method to weight the data types appropriately, i.e., to prioritize, filter and in some cases discard the data based on

how consistent it is with other data. The **integration** of multiple data sources in the inference serves two main purposes; to improve the biological significance of the GRN and to improve its ability to predict new experiments. The biological significance is almost certainly increased if the inference algorithm is fed with known facts about the biological system. However, to lower the prediction error, careful statistically sound model selection must be performed.

One important complication for GRN inference is the presence of non-Gaussian errors and outliers (Speed *et al.* 2000; Purdom *et al.* 2005). Most **Ordinary Differential Equations** (ODE) based algorithms having their origin in other fields assume Gaussian errors, e.g., least squares based algorithms. A good inference algorithm should be able to handle the great uncertainties connected with the large-scale experiments, which for mRNA data obtained from microarrays have been shown to closely follow a double exponential (Laplace) distribution (Purdom *et al.* 2005). At the same time, the algorithm should be able to handle more certain knowledge, obtained from experimental studies of smaller scale.

In this chapter we introduce and discuss a computational efficient inference procedure based on ODEs to infer GRNs treating all evidences as soft, and at the same time assuming a more realistic error distribution. We demonstrate its efficiency and the necessity of "softness" for a case study where we infer a network integrating the data at hand. Indeed, this network has high biological significance and furthermore low expected prediction error.

BACKGROUND

To determine causal relationships on a gene level we utilize a popular model class that is continuous in time and deterministic, see e.g., D'haeseleer *et al.* (1999), van Someren *et al.* (2000), Holter *et al.* (2001), Yeung *et al.* (2002), Segal (2003), Gustafsson *et al.* (2005), Wang *et al.* (2006) for some different settings. The restriction to deterministic models does not mean that the cell is assumed deterministic; however our assertions are deterministic and correspond to expectation values. Within this class of models, the rate of change of gene expression for gene *i*, $\dot{x}_i(t)$ can quite generally be described by

$$\dot{x}_i(t) = g_i(x_1(t), \cdots, x_N(t), t) + \varepsilon_i(t) \cdot$$

Here $x_j(t)$ is the gene expression at time *t* of gene *j*, *N* is the number of genes and $\varepsilon_i(t)$ a stochastic variable. However, such a model has infinitely many degrees of freedom and the model class must be restricted. One common choice is to limit g_i to time independent affine functions, *i.e.*, $\dot{x}_i(t) = a_0 + \sum_{j=1}^{N} w_{ij} x_j(t) + \varepsilon_i(t)$. In the case where the errors $\varepsilon_i(t)$ are individually independent following Gaussian distributions, minimizing the sum of squares is a proper choice for inference of the interaction matrix w.

The assumption of an affine (effectively linear) relationship is certainly wrong, but still there exist many reasons why it might be good enough. In the case of expression measurements from a single condition, we can consider it as the result of a linearization of the true non-linear function around a working point. This makes the approximation correct at least to a first order. It should also be noted that this formulation includes degradation into the model through negative self-interaction terms ($w_{ii} < 0$), and therefore all self-interactions are the sum of degradation and self-regulation. However, another common experimental

situation is that the observation may come from perturbations, which have the purpose of driving the system far away from its working point. We will therefore also investigate the use of some non-linear functions in the dynamical equations.

However, inference of linear functions has great computational advantages, which might be of crucial importance given the size of the system. Yet another reason for linear modelling is the lack of detailed knowledge of the "true" dynamical equations, due to the projection of gene regulation onto the space of genes only. The detailed equations governing the biochemical reactions can to some extent be described with Michaelis-Menten equations, but when projecting several such processes, the result is unclear.

Back to the inference problem again (with Gaussian error distribution) we might without loss of generality discard the intercept term a_0 from the model by centring data such that $\sum_k \dot{x}_i(t_k) = 0$. In the case where the number of measurements $K \geq N$, the matrix w can be obtained from solving *n* ordinary least squares problems, where *n* is the number of genes to be explained, i.e.,

$$w_{i\bullet} = \arg\min_{w_{i\bullet}} \sum_{k=1}^{K} \left(\dot{x}_i(t_k) - \sum_{j=1}^{N} w_{ij} x_j(t_k) \right)^2, \quad \forall i = 1,\ldots,n\,. \tag{1}$$

Here $w_{i\bullet} = \left\{ w_{i1} \ldots, w_{iN} \right\}$, t_k is the time instance where the measurement *k* has been carried out. Note also that we assume that the genes have been ordered such that the *n* genes which we aim at explaining are ordered first, followed by the rest explanatory *N-n* genes. The rationale for having $n < N$ is that the model should perform better for the genes associated to some specific process, but we still allow for novel regulators. However, in the typical microarray scenario, we have $K << N$, therefore we have infinitely many models that perfectly match the experimental data. To overcome this problem many different proposals have been made. Simulated experiments were suggested by D'haeseleer *et al.* (1999), choosing $w_{i\bullet}$ with the smallest L2 norm was exploited by Dewey *et al.* (2001) and sparseness was in the GRN inference setting incorporated by Yeung *et al.* (2002) and further extended by Wang *et al.* (2006). The use of a sparse solution relies on a biologically motivated assumption, i.e., each gene is only regulated by a few others while choosing $w_{i\bullet}$ with the smallest L2 norm corresponds to a fully connected network. However, working with perfect fits is a heavy assumption and perfect fits can be obtained from exactly *K* predictors for each gene, and as the number of experiments increases the network will become denser. A more stable version of utilizing sparseness, the LASSO, was proposed by Tibshirani (1996), which puts a L1 restriction on the predictors, i.e.,

$$\begin{aligned} w_{i\bullet} &= \arg\min \sum_{k=1}^{K} \left(\dot{x}_i(t_k) - \sum_{j=1}^{N} w_{ij} x_j(t_k) \right)^2, \quad \forall i = 1,\ldots,n \\ s.t. \quad & \sum_{j=1}^{N} \left| w_{ij} \right| \leq \Lambda_i. \end{aligned} \tag{2}$$

This was utilized by Gustafsson *et al.* (2005) for a global ad-hoc pick of $\Lambda_i = 0.1 \cdot \eta_i^{(2)}$, where $\eta_i^{(2)}$ is the smallest L2-norm among all the solutions $w_{i\bullet}$ with perfect fit. The network obtained from this study was shown to be biologically relevant, even in the extreme case where $n = N = 6178$ and *K*=73 for the yeast cell cycle. The L1 constraint induces the solution to be on the border of a hyper octahedron, with sharp corners at the coordinate axes, thus the minimum of the quadratic objective on the constraint

is therefore likely to be in one of the corners, where some coefficients are exactly zero. A computational strategy for tackling this problem more efficiently was proposed by Efron *et al.* (2004), LARS, which made it possible to pick Λ_i from several runs of the algorithm (see CV below). The effect of the inequality in (2) is both to regularize the solution and to select a subset of the predictors. In a statistical frame the shrink lowers the variance of the fit, which indeed is a stochastic variable since it is a fit of a single realisation of the data. The decreased variance in the fit comes from posing an additional property of the model, i.e., assuming the model parameters to be small in a L1 sense, which leads to an increased bias of the model.

The overall performance of the inference procedure can be determined by the *prediction error*, i.e., the expected error on new data. This is indeed a complicated measure since we expect the model to have different errors for different data, as the model only at best is a crude approximation of the reality. This fact together with the limited amount of experimental data makes systematic recycling of data necessary in order to estimate the expected prediction error as good as possible. Furthermore, the performance of the model is a combination of both the variance and the bias, thus if the model assumptions are good the performance of the inference is increased.

Another common approach to shrink $w_{i\bullet}$, thus lowering the $w_{i\bullet}$ variance is to modify the LASSO-inequality in (2) to yield $\sum_{j=1}^{N} {w_{ij}}^2 \leq \Omega_i$, which is called ridge regression (or Tikhonov regularization) and has been use for several decades in the statistician community (see, e.g., the text-book by Draper *et al.* 1998). Ridge regression and the LASSO relies on different model assumptions, the former assumes $w_{i\bullet}$ to be Gaussian distributed and none of the coefficients will almost certainly be exact zero, while the latter assumes sparse non-zero elements in $w_{i\bullet}$ (**Laplace distribution**, Hastie *et al.* 2001, discussed in *Incorporating* ***prior knowledge*** section)[1]. However, as a subset selection operator the L1-norm tends to be somewhat greedy and picks only the most correlated predictors. This might be good if we have the correct model and lots of data and particularly it is good for interpretability, but it will not lower the prediction error significantly (Zou *et al.* 2005). For GRN inference this obstacle might be severe for the prediction error, since many genes are correlated and the sparseity of data is tremendous. Nevertheless it might work well as a subset selection operator and in combination with other types of data. The strength of the LASSO approach for subset selection was observed in the DREAM *in silico* challenge (DREAM 2007), where the best algorithm for finding a directed unsigned description of the true network from an unknown model was based on the LASSO and adopted by Gustafsson *et al.* (In press a) with some nonlinear basis functions instead of x_i.

Within the LASSO framework (equation (2)) Zou *et al.* (2005) proposed another important extension, called the *elastic net*, which modifies the inequality for the LASSO and ridge regression to yield a combined L1 and L2 shrink constraint, with the computational benefits of the LARS algorithm (Zou *et al.* 2005):

$$\sum_{j=1}^{N} (1-\mu_i)\left|w_{ij}\right| + \mu_i \cdot {w_{ij}}^2 \leq \Lambda_i, \qquad (3)$$

where μ_i is a tuning parameter. The net effect of (3) is both a shrink of the number and the magnitude of the non-zero coefficients (if $\mu_i \in (0,1)$). Furthermore, it makes it possible to have more non-zero coefficients than experiments in the model. Particularly, (3) has a great impact when the regulatory genes are correlated; as mentioned above LASSO selects only the most correlated genes, while the elastic net

to some extent also includes slightly less correlated variables. In the case of highly correlated variables ridge regression and the elastic net have empirically been observed to yield lower prediction errors than the LASSO (see Zou *et al.* (2005) and references therein), but of course they include more indirect regulatory interactions as well. Therefore the choice of the mixture parameter μ_i for GRN inference may depend on whether direct interactions or low prediction error is preferred. For both the LASSO and elastic net the inequality regularizes the solution, increases the numerical stability, and decreases the variance of the $w_{i\bullet}$ estimates by an increased bias. However, as previously indicated, the model assumptions of all these approaches are very different. The elastic net is an important inference procedure for GRNs, especially when prediction of new experiments is desired we end the exploration of this procedure here. The strength of the elastic net for predicition was observed in the DREAM *gene expression* challenge (DREAM 2008), where the best performing algorithm in predicting new gene expression measurements was Gustafsson and Hörnquist, which utilized the elastic net (Gustafsson 2008). However, the main goal for inference is here for interpretational reasons, therefore we instead explore some extensions to the LASSO (equation (2)).

As mentioned above, the conventional method to infer a GRN by ODEs is to minimize the squared residual sum in (1). This is statistically motivated if the error distribution is Gaussian, since the parameters then correspond to the Maximum Likelihood Estimate. However, it has been argued by many researchers in the field that the microarray technique produce extremely noisy data with lots of outliers, i.e., the experiments have huge error bars (Speed et al. 2000; Ideker *et al.* 2001; Filkov *et al.* 2002; Purdom *et al.* 2005). Purdom *et al.* (2005) showed from empirical observations that the associated errors closely follow an asymmetric double exponential (**Laplace**) **distribution**., i.e., an asymmetric heavy tailed distribution. For such a distribution the **Least Absolute Deviation** (LAD) is preferred (Purdom *et al.* 2005), i.e., minimizing the internal fit in L1 sense. Some earlier attempts in this direction have been made (Yeung *et al.* 2002, Wang *et al.* 2006), however neither taking **prior knowledge** nor sparseness into account in a data driven fashion.

INCORPORATING PRIOR KNOWLEDGE

To get more reliable GRNs, **prior knowledge** or side knowledge must also be taken into account. An example of this may be the knowledge of binding sites of TFs in an upstream region to a gene, which indicates that the gene coding for the particular TF may regulate the corresponding gene. However, such type of knowledge often arises from other large-scale experiments, which also are connected with great uncertainties. We choose here to incorporate knowledge of that kind into the inequality constraint, i.e., we put less penalty on predictors suggested for regulation by other experiments than on those not suggested. In other words, we increase the probability of drawing regulator *j* of target *i* by imposing a low prior (Π_{ij}). By combining LAD, L1-minimziation and the prior knowledge, we obtain:

$$w_{i\bullet} = \arg\min \sum_{k=1}^{K} \left| \dot{x}_i(t_k) - \sum_{j=1}^{N} w_{ij} x_j(t_k) \right|, \quad \forall i = 1, \ldots, n$$
$$s.t. \quad \sum_{j=1}^{N} \Pi_{ij} \left| w_{ij} \right| \leq \Lambda_i. \tag{4}$$

Here $\Pi_{ij} \geq 0$ and contains a priori or side knowledge that gene *j* is a regulator to gene *i*, where 0 mean that we have maximum a priori belief in an interaction and a high value indicates belief in no interaction. This results in an ordered priority list of the regulators, which represents the order the regulators enter the model from an increase in Λ_i, i.e., we demand less evidence (correlation to the target) for the regulators for which we have other supporting evidence. The resulting effect is that many uncertain pieces are put together in a "soft evidence" based way, thus putting faith into **prior knowledge** still enabling for novel interaction discoveries. The prior may be a combination of different types of a priori knowledge, all associated with different information about the regulatory system. However, it is hard to manually put in how each type of **prior knowledge** contributes to the collected prior, Π_{ij}. To simplify the composition of different priors we introduce $P_{ij}^{m} \in [0,1]$, which represents the belief from prior type *m* that gene *j* is a regulator of gene *i*, unity represents maximum belief and zero means that we have no belief in such an interaction. We assume the prior to be a linear combination of all present **prior knowledge**, such that

$$\Pi_{ij} = \frac{\hat{\Pi}_{ij}}{\frac{1}{N}\sum_{j=1}^{N}\hat{\Pi}_{ij}},$$
$$\text{where} \quad \hat{\Pi}_{ij} = \max\left(1 - \sum_{m=1}^{npriors} \alpha_m P_{ij}^{\ m}, 0\right) \quad . \tag{5}$$

This particular formulation has the advantages that $\sum_{j=1}^{N} |\Pi_{ij}| = N$ and $\Pi_{ij} \geq 0$. This means that the norm of the priors are the same for all α_m for each gene, and that the left hand side of the inequality is a non-decreasing function of α_m. Note here that we introduce *npriors* extra parameters α_m, which reflect the overall relevance of the prior types to the inference problem. In the next section (*Model selection*) we discuss how these extra parameters can be determined.

Furthermore, since the use of the L1-inequality is to shrink the solution, we can impose $\Lambda_i = \lambda_i \cdot \eta_i^{(1)}$, where $0 \leq \lambda_i \leq 1$ and $\eta_i^{(1)}$ is the L1-norm of the solution with the smallest L1-norm among the perfect fits, i.e.,

$$\eta_i^{(1)} = \min \sum_{j=1}^{N} \left| w_{ij} \right|$$
$$s.t. \quad \sum_{k=1}^{K} \left| \dot{x}_i(t_k) - \sum_{j=1}^{N} w_{ij} x_j(t_k) \right| = 0. \tag{6}$$

This gives us a natural baseline for the magnitude of the constraint.

MODEL SELECTION

The above reverse engineering algorithm leaves some free parameters, $\bar{\lambda} = \left\{ \lambda_1, \ldots, \lambda_n \right\}$ and $\bar{\alpha} = \left\{ \alpha_1, \ldots, \alpha_{npriors} \right\}$, which should be tuned in a data driven fashion. Here we utilize the popular and easily interpretable model selection idea to minimize the leave-out error function (Cross-Validation, CV), which in the LAD setting for each gene is:

$$L_i(\lambda_i, \bar{\alpha}) = \sum_{k=1}^{K} \left| \dot{x}_i(t_k) - \sum_{j=1}^{N} w_{ij}^{t_k - out} x_j(t_k) \right|. \quad (7)$$

This is the same function as the objective, but on data unseen to the inference procedure. The coefficients $w_{ij}^{t_k - out}$ come from the solution to the regression problem where measurement *k* is excluded from the fit. In the microarray setting, the number of measurements, *K*, is indeed small, and to get as good fit as possible it is important that as few time points as possible are left out at the same time. Particularly, leave-one-out cross validation is preferred, but in practise it may take too long time to solve *K* optimization problems. Instead, more points may be left out at the same time, e.g., in the case study below we left 10% of the data out each time (ten-fold CV). Thus the problem we solve is an approximate minimization over the non-linear function $L(\bar{\lambda}, \bar{\alpha}) \equiv \frac{1}{n} \sum_i L_i(\lambda_i, \bar{\alpha})$ (see section *Implementation and computational considerations* for details).We stress the importance of proper model selection in this setting, since there is a substantial lack of data and numerous articles reporting on the huge errors present in the microarray technique (Ideker *et al.* 2001; Filkov *et al.* 2002; Purdom *et al.* 2005). This, together with other complicating issues, e.g., time-series being made from different cells, emphasize the need of careful model selection to avoid over-fitting.

Note that there are several alternatives based on the residual sum of squares to CV, e.g., AIC, BIC and GCV (Hastie *et al.* 2001, Thorsson *et al.* 2005). However, since we expect the model error to follow a non-Gaussian distribution the residual sum of squares is not a proper measure for the goodness of fit. Because of this, and the somewhat problematic estimation of degrees of freedom when including the prior distribution, we completely dismiss such an approach here.

GENERALIZATIONS

The model outlined in (4) has great computational advantages compared with a more general nonlinear model. However, two biologically motivated extensions taking into account some non-linear effects can be made within this framework. We introduce the non-linear functions f_j and g_i that should be specified in advance and are typically sigmoid functions, modelling saturation effects.

$$\dot{x}_i(t) = g_i \left(\sum_{j=1}^{N} w_{ij} f_j(x_j(t)) + \varepsilon_i(t) \right) \quad (8)$$

The problem is then converted to the standard form by applying the inverse function of g_i, g_i^{-1}, to (8) in which case the problem again becomes linear on the transformed data $g_i^{-1}(\dot{x}_i(t))$ and $f_j(x_j(t))$

, instead of $\dot{x}_i(t)$ and $x_j(t)$, respectively. Caution should be taken when choosing g_i, as g_i^{-1} must be well defined in the range of $\dot{x}_i(t)$ and not contain any flat regions. The nonlinear functions must be very restricted to avoid over-fitting and to be computationally efficient. An obvious choice is to let f_j and g_i to be sigmoid functions. For f_i this can be motivated on the regulator level basis, the mechanism being that expression level increments only affect the rate up to some total level, which may be due to the presence of supporting proteins to recruit the regulators (Ptashne *et al.* 2002). On the rate level, for g_i, it can be motivated by the mRNA creation speed, which may be limited by the presence of nucleotides, the speed of the polymerase and the degradation speed. The introduction of sigmoid functions formulates the need for a proper scale such that the non-linear effects can be observed where it still is possible for well-defined inverse transforms. Different parameter values may be optimal for different genes, but as we are normally short of data such a freedom will probably lead to an over-fit. In the sequel $\dot{x}_i(t)$ and $x_j(t)$ can be interpreted as $g_i^{-1}(\dot{x}_i(t))$ and $f_j(x_j(t))$ respectively.

Another generalization is to also include expression data sets obtained from other conditions (Wang *et al.* 2006), which we primarily do not aim to model. The sparseness of data is a strong motivation for doing this; however the primary regulations through TFs are condition specific, and different regulatory paths are active during different processes (Segal *et al.* 2003; Luscombe *et al.* 2004). Therefore, it can at most be taken into account as another "soft evidence", and should as such be put into the objective function in (4). Another important generalization when dealing with data from heterogeneous sources is to include external perturbations explicitly. For example, a cell culture may suddenly be warmed up to some temperature and remain there for a moment or be moved back to the original temperature causing different driving forces and different responses from the cell (Gasch *et al.* 2000).

Incorporating those generalizations the problem can be stated as to find the weighting of the mixture of prior belief parameters $\bar{\alpha}$ corresponding to the smallest leave-out error, i.e.,

$L^0 = \min_{0\le\bar{\alpha}\le 1} \frac{1}{n}\sum_{i=1}^{n} L_i(\bar{\alpha})$, where $L_i(\bar{\alpha}) = \min_{0\le\lambda_i\le 1} L_i(\bar{\alpha},\lambda_i)$ is the minimum leave-out error with respect to $L^0 = \min_{0\le\bar{\alpha}\le 1} \frac{1}{n}\sum_{i=1}^{n} L_i(\bar{\alpha})$ $L_i(\bar{\alpha}) = \min_{0\le\lambda_i\le 1} L_i(\bar{\alpha},\lambda_i)$ λ_i, where $L_i(\bar{\alpha}) = \min_{0\le\lambda_i\le 1} L_i(\bar{\alpha},\lambda_i)$ is the minimum leave-out error with respect to λ_i. Explicitly the leave-out error is

$$L_i(\bar{\alpha},\lambda_i) = \frac{1}{K}\sum_{k=1}^{K}\left|\dot{x}_i(t_k) - \sum_{j=1}^{N} w_{ij}^{t_k-out} x_j(t_k)\right|. \tag{9}$$

This corresponds to the mean absolute deviation of the prediction of the derivatives, where each term in (9) is the unbiased absolute deviation of the predictions (*i.e.*, excluding the estimated experiment from the actual inference). The coefficients $.w_{i\bullet}^{t_k-out}$ are thereby determined from solving $K\cdot n$ subproblems in which each minimization is carried out against a combination of the internal absolute deviation to the transformed data of interest and some other less important prior data set, explicitly

$$
\begin{aligned}
.w_{i\bullet}^{\,t_k-out} = \quad & \arg\min_{w_{ij}^{\,t_k-out}} \sum_{m\neq k} \left| \dot{x}_i(t_m) - \sum_{j=1}^{N} w_{ij}^{\,t_k-out} x_j(t_m) + u_i h_i(t_k) \right| + \\
& + \sum_{m=1}^{M} \alpha_{m+npriors} \sum_{p=1}^{K_m} \left| \dot{x}_i^{\,m}(t_p) - \sum_{j=1}^{N} w_{ij}^{\,t_k-out} x_j^{\,m}(t_p) + u_{n+p} h_{n+p}(t_p) \right| \\
s.t. \quad & \sum_{j=1}^{N} \Pi_{ij} \left(\left\{ \alpha_m \right\}_{m=1}^{npriors} \right) \left| w_{ij} \right| \leq \lambda_i \cdot \eta_i^{(1)}, \qquad (10)
\end{aligned}
$$

Here we have introduced *M* external expression data sets with K_m measurement microarrays, each having a driving force $h_{n+p}(t)$ known from the experimental setup, which for completeness also has been added to the primary expression data source.

IMPLEMENTATION AND COMPUTATIONAL CONSIDERATIONS

Now we present a strategy to make the implementation computationally tractable. First, we adopt the doubling variable strategy to code for the absolute values in the constraint. Every single absolute value in the equations is replaced by an addition of two non-negative parameters, e.g., $w_{ij} = a_{ij} - b_{ij},\ a_{ij}, b_{ij} \geq 0 \Rightarrow \left| w_{ij} \right| = a_{ij} + b_{ij}$, where a_{ij} represents positive regulation and b_{ij} negative regulation. This also introduces the opportunity to easily incorporate **prior knowledge** of activators and repressors, which may be important but not utilized here. Second, we perform the same "trick" on the Absolute Deviation. For clarity we drop the term regarding multiple expression sets and obtain:

$$
\left| \dot{x}_i(t_m) - \sum_{j=1}^{N} w_{ij}^{\,t_k-out} x_j(t_m) \right| = c_{ij}^{\,t_k-out} + d_{ij}^{\,t_k-out}. \qquad (11)
$$

The subproblems to be solved are then:

$$
\begin{aligned}
& \min \sum_{m\notin k} c_{ij}^{\,t_k-out} + d_{ij}^{\,t_k-out} \qquad , \forall k, i \\
& c_{ij}^{\,t_k-out} - d_{ij}^{\,t_k-out} = \dot{x}_i(t_m) - \sum_{j=1}^{N} \left(a_{ij}^{\,t_k-out} - b_{ij}^{\,t_k-out} \right) x_j(t_m), \qquad \forall m \neq k \\
& s.t. \quad \sum_{j=1}^{N} \Pi_{ij} (a_{ij}^{\,t_k-out} + b_{ij}^{\,t_k-out}) \leq \lambda_i \cdot \eta_i^1 \\
& \qquad a_{ij}^{\,t_k-out}, b_{ij}^{\,t_k-out}, c_{ij}^{\,t_k-out}, d_{ij}^{\,t_k-out} \geq 0 \qquad \forall j, m \neq k \qquad (12)
\end{aligned}
$$

Thus the $K \cdot n$ subproblems that should be solved for each parameter set $(\overline{\lambda}, \overline{\alpha})$ are all LP problems each consisting of $2 \cdot (K + N - K^{out})$ variables and $(K - K^{out} + 1)$ constraints with non-negative variables, where the number K^{out} corresponds to the number of experiments left out from the inference. For clarity we assume the number of left out experiments $K^{out} = 1$ in (12) above, but the generalization is straightforward. As K<<N the substitutions are not that devastating for the implementation, since we can solve the subproblems using the simplex method for **Linear Programming**. The full optimization problem now consists of a hierarchy of subproblems,

$$L^0 = \min_{0\le\bar{\alpha}\le 1} \quad \frac{1}{n}\sum_{i=1}^{n} L_i(\bar{\alpha})$$
$$L_i(\bar{\alpha}) = \min_{0\le\lambda_i\le 1} \frac{1}{K}\sum_{k=1}^{K}\left|\dot{x}_i(t_k) - \sum_{j=1}^{N}\left(a_{ij}^{\,t_k-out} - b_{ij}^{\,t_k-out}\right) x_j(t_k)\right|, \tag{13}$$

where $a_{ij}^{\,t_k-out}, b_{ij}^{\,t_k-out}$ are solutions to (12). In the optimization over the parameters $(\bar{\lambda},\bar{\alpha})$, we search for each $\bar{\alpha}$ the minimizer $\bar{\lambda}^*$ by solving $K\cdot n / K^{out}$ LP-problems. However, whenever we have solved a problem for the *n* genes and are modifying the left out experiments, or $\bar{\lambda}$, or even $\bar{\alpha}$, we may use the previously found solution as seed for the next optimization as a warm start for the optimization engine, which greatly decreases the calculation time. Taking this into account we implement the algorithm as follows. First, we set $\bar{\alpha}=0$, $\bar{\lambda}=0$ and for each *i* starting from $\lambda_i = 0$ we have the trivial solution $w_{i\bullet}^{\,t_k-out}=0$ for all *i*, *k*. Then increasing λ_i by 0.01 we can assume that we have a solution nearby and use the previously obtained $w_{i\bullet}^{\,t_k-out}$ (or $a_{ij}^{\,t_k-out}, b_{ij}^{\,t_k-out}$ in the implementation) as initial guesses to warm start the LP-solver. We follow a particular starting solution, instead of utilizing restarts. This saves computation time, but may lead us wrong in some (hopefully rare) cases. The smoothness of the found solutions indicate however that this is a minor problem. In practice, we start from $w_{i\bullet}^{\,t_k-out}=0$ and therefore prefer solutions near to this starting solution. Furthermore, as we have solved the problem for some leave out experiments at a particular λ_i, we can switch to use the solution obtained for the same λ_i (but for different *k*) as initial guess, which decrease the computation time further. As more leave-out calculations are performed the initial guess gets closer to the solution, particularly we then choose as an initial guess the median values of the calculated $a_{ij}^{\,t_k-out}, b_{ij}^{\,t_k-out}$ at earlier *k*, and estimated $c_{ij}^{\,t_k-out}, d_{ij}^{\,t_k-out}$ using the same technique.

In practise, the calculations are carried out by using the CPLEX primal LP-solver disabling the presolver and incorporating the initial values by MatLab implementation of Giorgetti (2005) to call the CPLEX-solver.

CASE STUDIE – YEAST CELL CYCLE

To exploit what type of nonlinearities and **prior knowledge** may in practise help the understanding of gene regulatory networks and compare with other approaches, we utilize the previously often explored extended Spellman dataset (Spellman *et al.* 1998; Cho *et al.* 1998). It consists of 4 time-series of measured mRNA-levels during one or more periods of the cell cycle with different synchronization processes presented as log-ratios. In the pre-processing of the data we filled in missing values with the KNNimpute (Troyanskaya *et al.* 2001) algorithm and estimated the derivatives using forward differences, i.e., $\dot{x}_i(t_k) = \dfrac{x_i(t_{k+1}) - x_i(t_k)}{t_{k+1} - t_k}$. The $x_i(t_k)$ is normalized such that each gene has zero mean and unit standard deviation. This leads to a dataset with *K*=69 experiments and *N*=4153 Verified ORFs (VORFs) (Fisk *et al.* 2006), which we take all as presumptive regulators. However, since we work with cell cycle data we only expect those genes associated to the cell cycle to be explainable, therefore we pick only those

n=420 annotated as associated to the cell cycle by the Gene Ontology (GO) (Ashburner *et al.* 2000, Fisk *et al.* 2006)[2] to serve as the genes whose transcription rates we predict.

To explore whether nonlinear functions should be utilized in this case we tested whether f_j and g_i as arctan functions can lower the leave-out errors $L(\overline{\lambda},\overline{\alpha})$ compared with the linear model. We choose $f_j(x)=\tan^{-1}(x),\forall j$ and g_i, such that $g_i^{-1}\left(\dot{x}\right)=\tan\left(b_i\left(\dot{x}-a_i\right)\right),\forall i$ where $a_i=\frac{1}{K}\sum_{k=1}^{K}\dot{x}_i(t_k)$ and $b_i=\frac{\pi}{3\cdot\max_k\left|\dot{x}_i(t_k)-a_i\right|}$. The scaling ensures that the domain of g_i^{-1} is centred on zero and that the maximum value of $\dot{x}(t)$ is attained within the domain of g_i^{-1}, which is not too flat and accounts for significant non-linear effects. However, from the table below (Table 1) we see that the simple linear model outperforms the others in this case.

Hence, in the forthcoming we discard the nonlinear functions from the analysis and focus on the incorporation of **prior knowledge**. The discussion in the introduction leads us to test the influence of the following types of prior/additional knowledge (Table 2):

Table 1.

Base functions	$\dot{x}$ and x	$\dot{x}$ and $f(x)$	$g^{-1}(\dot{x})$ and x	$g^{-1}(\dot{x})$ and $f(x)$
$L(\overline{\alpha}=0)$	0.0244	0.0278	0.0249	0.0253

Table 2.

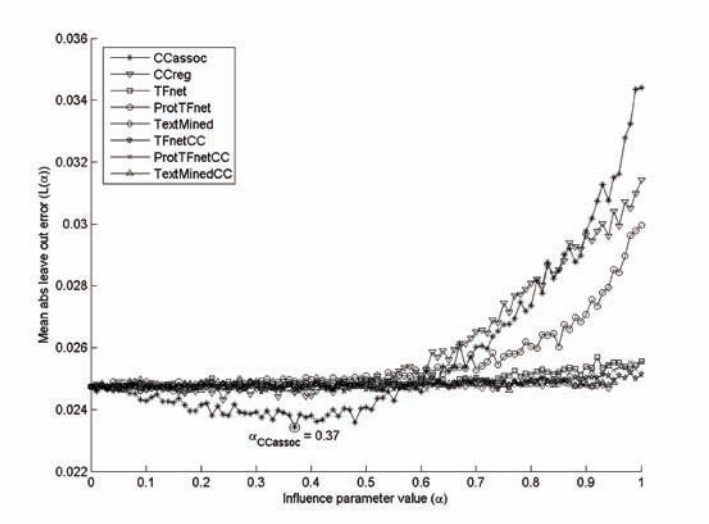
Short name	Source	Type	Statistics of number of regulators #{ $P_{i\bullet}^m>0$ }		
			Average	Min	Max
CCassoc	GO:7049	$P_{ij}^m\in\{0,1\}$	420	420	420
CCreg	GO:51726	$P_{ij}^m\in\{0,1\}$	154	154	154
TFnet	Yeastract	$P_{ij}^m\in\{0,1\}$	4.1	0	22
ProtTFnet	Yeastract + DIP	$P_{ij}^m\in\{0,1\}$	21	0	96
TextMined	GeneLinks/PubMed	$P_{ij}^m\in[0,1]$	1877	0	4027
TFnetCC	TFnet/CCassoc	$P_{ij}^m\in\{0,1\}$	1.25	0	8
ProtTFnetCC	ProtTFnet/CCassoc	$P_{ij}^m\in\{0,1\}$	4.91	0	22
TextMinedCC	TextMined/CCassoc	$P_{ij}^m\in[0,1]$	207	0	417

- Cell cycle association, CCassoc (GO: 7049); the genes previously reported as associated to the cell cycle should be the ones most active and their regulations most detectable. This will be a binary variable with 420 regulators for each gene.
- Cell cycle regulation, CCreg (GO: 51726); the genes annotated as regulators of the cell cycle should have an impact on the regulation of other genes. This will be a binary variable with 154 regulators for each gene.
- Transcription Factor binding edges, TFnet. This is a binary matrix with about 4.1 interactions per gene, downloaded from YeastTract (Teixeira *et al.* 2006). The full TF-network obtained form the database has 163 known TFs which bind to any of the 4315 VORFs, for which 132 of them are TFs suggested to bind to targets associated to the cell cycle.
- TF binding proteins, ProtTFnet; From the DIP yeast core protein interaction network (Xenarios *et al.* 2000) and the full TF-network (Teixeira *et al.* 2006) we project TF-target regulations onto proteins that bind to TFs. The rationale being that TF binding proteins and TFs may form dimers which in turn regulate their targets. As TF-levels often are low and hard to detect we represent the levels of such dimers by the expression of the TF binding protein.
- Text mined associations, TextMined; From the PubMed (2007) archive we derived an undirected network of gene associations. We associate two genes to each other if the genes where reported in the same articles in PubMed from Gene Links, with the negative logarithm of the P-value for randomly retrieving that number of associations (Lundström 2007). The idea behind this network is that genes associated to each other from similar publications may be interrelated. These numbers are divided by the maximum numbers, making unity the largest value. For implementation reasons we only store these numbers using two decimals.
- Intersections of TFnet, ProtTFnet and TextMined with the cell cycle association prior accounts for the condition specific counterparts TFnetCC, ProtTFnetCC and TextMinedCC respectively. The rationale is to filter out edges active in other conditions, CCassoc is preferred to CCreg because of its semi-dense nature.

To explore the weighting among the prior information we bring them in one at the time, i.e., leaving the others out from the inference. For each type of prior to be incorporated, we perform a one dimensional grid search using step size of 0.01 from 0 to 1 of the corresponding parameter, motivated by the assumption that many priors do not improve the GRN inference at all. The strategy is that this might rule out some priors and gives a hint of the optimal parameter setting. We see the result of these searches in Figure 1, where it is clear that CCassoc is the most important **prior knowledge**. Evidently, it serves as a good compromise between excluding many non-regulators and preserving indirect or previously undetected regulations. The optimal value of $\alpha_{CCassoc} = 0.37$ being in the intermediate regime stresses the importance of using the prior as "soft evidence", thus enabling for novel interaction findings while it still gives faith to earlier findings. It should also be noted that in all cases it is a less useful idea to fully rely on the **prior knowledge** and particularly in the CCassoc case it is devastating, leading to the poorest fits among all.

The next step is to set $\alpha_{CCassoc} = 0.37,$ and we perform a one dimensional grid search for the other kinds of **prior knowledge** to explore what other priors produce the most information in pair with CCassoc. The result is visualized in Figure 2. Strikingly, no improvement is evident in this search, and therefore we conclude that no more prior should be incorporated.

Figure 1. The leave-out function, $L(\bar{\alpha})$, displayed as a function of the influence parameter $\bar{\alpha}$. As we can see it is only the CCassoc that lowers $L(\bar{\alpha})$ detectably and the minimizer $\alpha_{CCassoc} = 0.37$ has a function value of 0.0234.

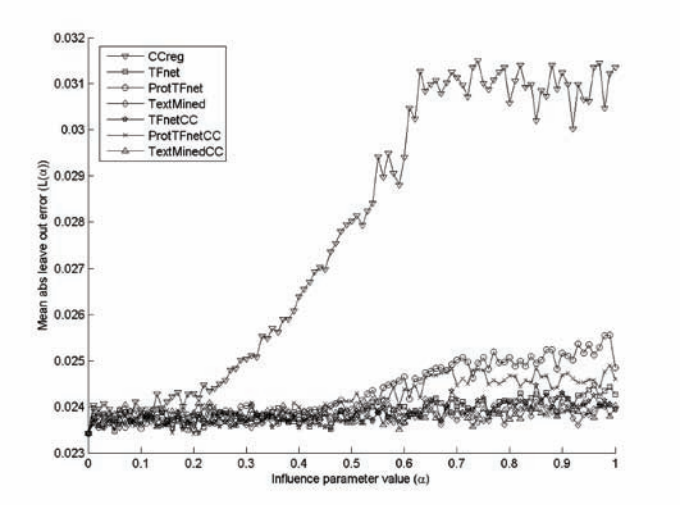


To compare the method against earlier attempts we performed the LASSO approach to regress in a least square sense (equation (2)). Here, LASSO was implemented by vanden Berghen (2007) by utilizing the LARS strategy from Hastie *et al.* 2004 and does not incorporate **prior knowledge**[3]. The LASSO net consisted of 2704 edges (27% with a positive sign) whereas the **Least Absolute Deviation** (LAD) network contained 24620 edges (whereof 43% positive). Among those we observe 1472 similar edges of which 1470 (99.9%) had the same signs in the two networks. Thus, one main effect of the LAD objective is an increase in the number of predictors, which in this case roughly coincide with the maximum number of edges $(K - K^{out}) \cdot n \approx 26000$. Hence, the average in-degree is about 62, which is far less than a complete network has. Still, though the magnitude of the L1-shrink is low.

In case one wants to prune the network, one can utilize the common rule of thumb of choosing the smallest parameter λ_i corresponding to at most one standard deviation of the prediction error above the minimum (Hastie *et al.* 2001). This might be more important here than in the more common least squares case due to the flatness of the LAD error function in (7). A flat error function means that many values of the error function have indistinguishable function values when adding noise. Thus, due to multiple testing, the observed minimizer is biased towards the middle of the testing interval of Λ_i, which probably means a denser network.

Finally, we note the **prior knowledge** incorporation in the LAD setting resulted in the optimal network corresponding to $\alpha_{CCassoc} = 0.37$ (all other $\alpha_i = 0$), which can be seen as a compromise between the CCassoc prior and the expression data. Thus 51.9% (13162) of the edges have a cell cycle associated gene as a regulator and 24.5% (6210) of the edges overlap with the network corresponding to all $\alpha_i = 0$, leaving 31.8% (8062) not recognised in any of the former two. Evidently, this procedure makes it possible to discover novel edges and at the same time preserve edges obtained from multiple sources.

Figure 2. The leave-out function, $L(\bar{\alpha})$, displayed as a function of the influence parameter $\bar{\alpha}$, when $\alpha_{CCassoc} = 0.37$ is fixed. No lower value of the leave-out function can be obtained by including more **prior knowledge**.



CONCLUSION

We have outlined a computational efficient optimization approach to the reverse engineering problem of gene regulatory networks, including a careful model selection. Furthermore, we have performed a case study on a genome-wide scale to demonstrate its practical applicability. The presented methodology is able to take into account various types of **prior knowledge** to overcome the data deficiency problem. The **prior knowledge** is incorporated at several stages, from the usage of double L1 regression to the **integration** of multiple large-scale data sets. The use of a L1 restriction on the predictor coefficients origins in the common belief that GRNs have a sparse nature, while the usage of the **least absolute deviation**, LAD, in place of least squares, as objective function, arise from the finding that the error distribution follows a double exponential (**Laplace**) form. However, the most important contribution of this chapter is to introduce a "data driven" incorporation of multiple data sets, where each data set only is included to the extent it decreases the expected prediction error, which is found from **cross-validation** procedures.

For the case study, we explored data from the yeast cell cycle. Interestingly, we observed that the ability of being associated to the cell cycle is by itself the single most important **prior knowledge** to incorporate. However, it is evident that the preference of these regulators should not be complete, since we observe intermediate incorporation values to be optimal. This emphasizes the need of using "soft" **prior knowledge**, contrary to some earlier works (e.g. Luscombe *et al.* (2004)). Indeed, the networks corresponding to indiscriminately incorporation of **prior knowledge** exhibited the largest prediction errors, even worse than no prior information at all. Eventually, in the incorporation steps, we found the optimal network to rely partially on genes annotated to the cell cycle as regulators. This network has many edges similar to its sources, but remarkably includes also a substantial portion (31.8%) of novel edges.

Furthermore, we incorporated non-linearities as sigmoid basis functions into the model, which at the end turned out to be neglected in our case study. One interpretation of this result may be that the linear functions are fine for single conditions and non-linearities will be of more importance when incorporating more heterogeneous data sets. This may be, since a linearization of a non-linear function can work accurately for a particular condition, but differently for combined conditions. Hence, at the same time when considering the effects from various conditions it may be of greater importance to work with non-linear models.

FUTURE RESEARCH DIRECTIONS

The whole field of reverse engineering of biological networks still suffers from the absence of common measures for comparing different algorithms. Some attempts, though, have been made to arrange objective competitions, e.g., the DREAM (2007) initiative. However, if these competitions present real data, there are the obvious risks both that data are recognized and that the true network is too poorly understood (containing both false positives and false negatives) to act as a gold standard. On the other hand, an in silico network might be too unrealistic, possibly missing features evolutionary developed but yet not recognized by researchers, and certainly not taking into account all well known features of gene regulation. Nevertheless, a very important step forward would be to have a common standard against which most of the research community agrees to assess their algorithms.

The methodology presented here is promising for reverse engineering of large-scale GRNs, but there are several possibilities for improvements. One such future research direction comes from the observation that even though we use the LP simplex method and start from nearby solutions, the optimization takes time. Especially, the local optimization procedure with respect to λ_i is time-consuming and not always satisfactory. Therefore, fast procedures to find optimal λ_i taking into account the specific problem structure -as the LARS algorithm does for LASSO type problems- would drastically increase the performance of the algorithm. To take into account as much data as possible, a mixture of weighted L1 and L2 norms is desired both in the objective function and in the regularizer. Another important property may be the ability to use warm start strategies both for tuning the mixture of the priors, and to speed up Cross Validation. A step into this direction is the R-package by Friedman *et al.* 2008 which enable the user to follow a solution path when increasing the constraint for a mixture of weighted L1 and L2 norms this procedure within the realm of weighted least squares.

Yet, another direction concerns the lack of improvement for most of the **prior knowledge**. Whether it depends on the type of prior incorporated, low quality data (the Spellman data set is known to be of limited quality (see e.g. Filkov *et al.* 2002) and is rather old) or something else, is an important issue, but beyond the scope of the present text. It is often assumed that increased structural information, such as the interaction type **prior knowledge**, leads to better reconstruction of the networks. Here, we can in the case study observe the converse for most types of prior information. It also brings up the important issue of how the error function should be constructed. In this chapter, we only relied on the microarray data to estimate the error, but this is somewhat problematic since we know that these data are extremely noisy, and if we believe in other sources they could possibly stabilize the error function. How this should be performed is an important still open question.

Finally we mention it would be of interest to explore how non-linearities and multiple conditions could improve the reverse engineering of the network and if the **prior knowledge** can be refined in a biologically motivated way to influence the final model. All this together would greatly increase the quality of the inferred networks and thus make them more tractable for the goal of understanding the biology on a systems level.

REFERENCES

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., et al. (2000). Gene ontology: Tool for the unification of biology. *Naure Genetics*, 25, 25–29. Annotations retrieved in December 2007, from http://www.yeastgenome.org/

Barabasi, A. L., & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews. Genetics*, *5*, 101–113. doi:10.1038/nrg1272

Bonneau, R., Reiss, D. J., Shannon, P., Hood, L., Baliga, N. S., & Thorsson, V. (2006). The inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*(5), R36. doi:10.1186/gb-2006-7-5-r36

Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene networks: How to put the function in genomics. *Trends in Biotechnology*, *20*, 467–472. doi:10.1016/S0167-7799(02)02053-X

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., & Wodicka, L. (1998). A genomewide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, *2*, 65–73. doi:10.1016/S1097-2765(00)80114-8

D'haeseleer. P., Wen, X., Fuhrman, S., & Somogyi, R. (1999). Linear modeling of mRNA expression levels during CNS development and injury. In R. B. Altman, A. K. Dunker, L. Hunter, T. E. Klein & K. Lauderdaule (Eds.), *Pacific Symposium on Biocomputing*, 4, 41–52. Singapore: World Scientific Publishing Co.

Draper, N. R., & Smith, H. (1998). *Applied regression analysis*, 3rd ed. New York: Wiley.

DREAM. Dialogue on Reverse-Engineering Assessment and Methods. (2007). Retrieved in January 2009, from http://wiki.c2b2.columbia.edu/dream/index.php/The_DREAM_Project

DREAM. Dialogue on Reverse-Engineering Assessment and Methods. (2008). Descriptions of the challenges. Retrieved in January 2009, from http://wiki.c2b2.columbia.edu/dream/index.php/The_DREAM3_Challenges

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, *32*, 407–499. doi:10.1214/009053604000000067

Filkov, V., Skiena, S., & Zhi, J. (2002). Analysis techniques for microarray time-series data. *Journal of Computational Biology*, *9*, 317–330. doi:10.1089/10665270252935485

Fisk, D. G., Ball, C. A., Dolinski, K., Engel, S. R., Hong, E. L., & Issel-Tarver, L. (2006). Saccharomyces cerevisiae S288C genome annotation: A working hypothesis. [from http://www.yeastgenome.org/]. *Yeast (Chichester, England)*, *23*(12), 857–865. Retrieved in December 2007. doi:10.1002/yea.1400

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Lasso and elastic-net regularized generalized linear models. Retrieved from http://www-stat.stanford.edu/~hastie/Papers/glmnet.pdf

Gasch, A. P., Spellman, P. T., Kao, C. M., Cramel-Harel, O., Eisen, M. B., & Storz, G. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, *11*, 4241–4257.

Giorgetti, N. (2005). Matlab MEX interface for the CPLEX callable library. Retrieved in December 2007, from http://www.dii.unisi.it/~hybrid/tools/mex/downloads.html CPLEX version 10 retrieved from http://www.ilog.com/ (commercial software)

Gustafsson, M., Hörnquist, M., Björkegren, J., & Tegnér, J. (in press). Genomewide system analysis reveals stable yet flexible network dynamics in yeast. *IET Systems Biology*.

Gustafsson, M., Hörnquist, M., & Lombardi, A. (2005). Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*, 254–261. doi:10.1109/TCBB.2005.35

Gustafssson, M., & Hörnquist, M. (2008). Gene expression by the elastic net. In M. Kellis, A. Califano & G. Stolovitzky (Eds.), *DREAM3, RECOMB satellite proceedings* (p. 48 and p. 133). Boston, MA.

Gustafssson, M., & Hörnquist, M. (in press). Reverse engineering of gene networks with LASSO and nonlinear basis functions. *Annals of the New York Academy of Sciences*.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer-Verlag.

Holter, N. S., Maritan, A., Cieplak, M., Fedoroff, N. V., & Banavar, J. R. (2001). Dynamical modelling of gene expression data. *Proceedings of the National Academy of Sciences of the United States of America*, *98*, 1693–1698. doi:10.1073/pnas.98.4.1693

Ideker, T., Thorsson, V., Ranish, J. A., Christmas, R., Buhler, J., & Eng, J. K. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, *292*(5518), 929–934. doi:10.1126/science.292.5518.929

Lundström, J. (2007). *Private communication*.

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A., & Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, *431*, 308–312. doi:10.1038/nature02782

Ma'ayan, A., Lipshtat, A., Iyengar, R., & Sontag, E. D. (2008). Proximity of intracellular regulatory networks to monotone systems. *IET Systems Biology*, *2*, 103–112. doi:10.1049/iet-syb:20070036

Marguerat, S., Jensen, T. S., de Lichtenberg, U., Wilhelm, B. T., Jensen, L. J., & Bähler, J. (2006). The more the merrier: Comparative analysis of microarray studies on cell cycle-regulated genes in fission yeast. *Yeast (Chichester, England)*, *23*, 261–277. doi:10.1002/yea.1351

Milo, R., Shen-Orr, S., Itzkovitz, N., Kashtan, D., Chklovskii, D., & Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, *298*, 824–827. doi:10.1126/science.298.5594.824

Ptashne, M., & Gann, A. (2002). *Genes and signals*. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press.

PubMed. (2007). Retrieved in December 2007, from http://www.ncbi.nlm.nih.gov/sites/entrez/

Purdom, E., & Holmes, S. P. (2005). Error distribution for gene expression data. *Statistical Applications in Genetics and Molecular Biology*, *4*(1), 16. doi:10.2202/1544-6115.1070

Schadt, E. E., Lamb, J., Yang, X., Zhu, J., Edwards, S., & Guhathakurta, D. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, *37*, 710–717. doi:10.1038/ng1589

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., & Friedman, N. (2003). Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, *34*(2), 166–176. doi:10.1038/ng1165

Segal, M. R. (2003). Regression approaches for microarray data analysis. *Journal of Computational Biology*, *10*(6), 961–980. doi:10.1089/106652703322756177

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., & Eisen, M. B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, *9*, 3273–3297.

Teixeira, M. C., Monteiro, P., Jain, P., Tenreiro, S., Fernandes, A. R., Mira, N. P., et al. (2006). The YEASTRACT database: A tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*, *Nucleic Acids Research*, 34, D446-D451. Oxford University Press. Retrieved in December 2007, from http://www.yeastract.com

Thorsson, V., Hörnquist, M., Siegel, A. F., & Hood, L. (2005). Reverse engineering galactose regulation in yeast through model selection. *Statistical Applications in Genomics*, 4(1).

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Methodological*, *58*, 267–288.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., & Tibshirani, R. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics (Oxford, England)*, *17*(6), 520–525. doi:10.1093/bioinformatics/17.6.520

van Someren, E. P., Wessels, E., Backer, L. F. A., & Reinders, M. J. T. (2003). Multicriterion optimization for genetic network modelling. *Signal Processing*, *83*, 763–775. doi:10.1016/S0165-1684(02)00473-5

vanden Berghen, F. (2007). The new ultraFast LARS engine with n-fold-cross-validation and ridge regression. Retrieved in December 2007, from http://www.applied-mathematics.net/

Wang, Y., Trupti, J., Zhang, X., Xu, D., & Chen, L. (2006). Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics (Oxford, England)*, *22*, 2413–2420. doi:10.1093/bioinformatics/btl396

Xenarios, I., Rice, D. W., Salwinski, L., Baron, M. K., Marcotte, E. M., & Eisenberg, D. (2000). DIP: The database of interacting proteins. [from http://dip.doe-mbi.ucla.edu/]. *Nucleic Acids Research*, *28*, 289–291. Retrieved in December 2007. doi:10.1093/nar/28.1.289

Yeung, M. K. S., Tegnér, J., & Collins, J. J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, *99*, 6163–6168. doi:10.1073/pnas.092576199

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B. Methodological*, *67*(2), 301–320. doi:10.1111/j.1467-9868.2005.00503.x

ADDITIONAL READING

Bonneau, R., Reiss, D. J., Shannon, P., Hood, L., Baliga, N. S., & Thorsson, V. (2006). The Inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*(5), R36. doi:10.1186/gb-2006-7-5-r36

Gustafsson, M., Hörnquist, M., & Lombardi, A. (2005). Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *2*, 254–261. doi:10.1109/TCBB.2005.35

Gustafssson, M., & Hörnquist, M. (in press). Reverse engineering of gene networks with LASSO and nonlinear basis functions. [accepted for publication]. *Annals of the New York Academy of Sciences*.

Margolin, A. A., & Califano, A. (2007). Theory and limitations of genetic network inference from microarray data. *Annals of the New York Academy of Sciences*, *1115*, 51–72. doi:10.1196/annals.1407.019

Segal, M. R. (2003). Regression approaches for microarray data analysis. *Journal of Computational Biology*, *10*(6), 961–980. doi:10.1089/106652703322756177

Stolovitzky, G. A., Monroe, D., & Califano, A. (2007). Dialogue on reverse engineering assessment and methods: The DREAM of high throughput pathway inference. *Annals of the New York Academy of Sciences*, *1115*, 1–22. doi:10.1196/annals.1407.021

Wang, Y., Trupti, J., Zhang, X., Xu, D., & Chen, L. (2006). Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics (Oxford, England)*, *22*, 2413–2420. doi:10.1093/bioinformatics/btl396

Yeung, M. K. S., Tegnér, J., & Collins, J. J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, *99*, 6163–6168. doi:10.1073/pnas.092576199

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *JRSSB*, *67*(2), 301–320. doi:10.1111/j.1467-9868.2005.00503.x

KEY TERMS AND DEFINITIONS

Data Integration: Is the merging of data stemming from different sources, such as expression data and TF-binding data.

Prior Knowledge: is our prior belief of a certain event. In this chapter we fuse different pieces of e.g. structural data into our prior belief, which enables the integration of structural and expression data.

Soft Evidence: is the concept to take into account multiple pieces of evidence as uncertain knowledge. We use the concept to stress the fact that we are using the multiple prior edge information to increase the probability for an edge, and not merely as filters.

Least Absolute Deviation (LAD): is here the minimization criteria which we base our solutions on. It is known to be more robust towards outliers than the more popular least squares method.

Sparseness: in a regulatory network context means that there are relatively few interactions per gene.

Linear Programming (LP): denotes the optimization problem where the objective function is linear and there are linear constraints. Efficient optimization algorithms for solving LP problems exist, especially the simplex method.

Warm Start: Optimization is a starting of the optimization algorithm in a state where it is close to the optimum.

ENDNOTES

1 Note that the distribution of the model parameters is in a Bayesian notion stochastic, hence introduces the regularization a prior distribution of the parameters. A non-Bayesian notion simply indicates that our regularizer is based on the assumption that the model parameters follow a particular distribution.

2 By associated to the cell cycle we mean annotated to a biological process term with the relation "is_a" or "is_part_of" the GO-term 7049. The cell cycle annotation study is still an ongoing research project (Marguerat *et al.* 2006), and GO has the intention to be an updated data source.

3 Recently Friedman *et. al.* (2008) implemented the elastic net combined with weighting of priors and weighted regression, which also enables the outlined methodology within a quadratic objective.

Section 7

Network Simulation Studies

Chapter 21
Dynamic Links and Evolutionary History in Simulated Gene Regulatory Networks

T. Steiner
Honda Research Institute Europe GmbH, Germany

Y. Jin
Honda Research Institute Europe GmbH, Germany

L. Schramm
Technische Universitaet Darmstadt, Germany

B. Sendhoff
Honda Research Institute Europe GmbH, Germany

ABSTRACT

In this chapter, we describe the use of evolutionary methods for the in silico generation of artificial gene regulatory networks (GRNs). These usually serve as models for biological networks and can be used for enhancing analysis methods in biology. We clarify our motivation in adopting this strategy by showing the importance of detailed knowledge of all processes, especially the regulatory dynamics of interactions undertaken during gene expression. To illustrate how such a methodology works, two different approaches to the evolution of small-scale GRNs with specified functions, are briefly reviewed and discussed. Thereafter, we present an approach to evolve medium sized GRNs with the ability to produce stable multi-cellular growth. The computational method employed allows for a detailed analysis of the dynamics of the GRNs as well as their evolution. We have observed the emergence of negative feedback during the evolutionary process, and we suggest its implication to the mutational robustness of the regulatory network which is further supported by evidence observed in additional experiments.

DOI: 10.4018/978-1-60566-685-3.ch021

INTRODUCTION

In biology, organisms consist of large numbers of heterogeneous elements existing on many spatial scales that nonlinearly interact physically and chemically on various timescales. Such interactions are the result of natural selection, the outcome of evolutionary process as driven by genetic variation and environmental change. It is one of the aims of systems biology to understand the principles of these interactions holistically and thereby to clarify the relationship between the microscopic regulatory dynamics and the macroscopic phenotypic properties of organisms. Knowledge from evolutionary history alone is insufficient for this endeavor because on the one hand it is not available in sufficient detail, and on the other, we can only infer the dynamics from extant organisms. Consequently, our knowledge about evolutionary lineage with regard to the dynamic properties of organisms is principally incomplete. Computer simulations and especially simulations of the evolutionary development of organisms provide us with a powerful tool to address this problem. Although having the inherent drawback of substantial simplifications of the processes involved, computer simulations of development offer us the possibility to study the complete dynamics that evolve during the artificial phylogenetic and ontogenetic history of organisms. Furthermore, in addition to studying known biological systems and processes, we can also study possible alternatives that – at least to date – we do not see in nature, which has been nicely phrased as studying "life as it could be" (Forbes (2000)). Also the seemingly apparent drawback of simplification can actually help us to not get lost in too much biological detail and therefore allows us to see the broader "systems" picture more clearly.

Therefore, we should regard computer simulations as powerful tools to investigate facts that are not available from analysis of biological data alone. Of course, the connection between the computational model and the real biological system, i.e., the abstraction level of the model, needs to be taken into careful account when interpreting the results. For example, in a computational model of evolutionary development, it is possible to observe all dynamics that take place on a simulated GRN, and at the same time all simulated GRNs can be put into their evolutionary context. This is possible because the data of such an experiment is at the same time complete and limited with regard to its complexity, making it possible to perform a thorough analysis. However, these observations are coupled to the simulation environment and it will be very unlikely that the simulated processes will fully mirror those that evolved in nature. What can be found however, are principles, for example, the role of feedback. Possible reasons for the emergence of such principles can then be carefully deduced from the computational models.

This chapter will review approaches to the simulation of the evolution of GRNs for systems biology, and will present a method for the simulation of evolutionary multicellular development. We will show how models are chosen in a task specific manner, such as evolving GRNs for certain behaviors like cellular clocks/oscillators. We then discuss the scientific value of these approaches.

The chapter is organized as follows: Following this introduction, we will briefly describe standard methods in biology used to collect data about GRNs from organisms and how knowledge from the data is extracted. We will then discuss the limitations of these approaches to biological research in general and thereby elucidate the underlying motivation in using computational models, emphasizing especially the expected benefits. After a review of different models used for simulated evolution of GRNs, we will describe a model of evolutionary development that we focus on in our own research, with an emphasis on the choice of abstraction level. An analysis made on evolutionary runs yielded from this model producing stable cell growth, is given as an example. The importance of understanding the features of evolved individuals in terms of both the dynamic structure of GRNs and their evolutionary history will be highlighted.

Finally, we will discuss the use of computational models of evolutionary development in biological research as an important approach to understand the systems-aspect (e.g. Kitano (2002)) and point out possible future research directions.

RECONSTRUCTING GENE REGULATORY NETWORKS FROM BIOLOGICAL DATA

Introduction

If we want to infer and assess GRNs for computational analysis from biologic data, we need background knowledge about gene regulation and the common methods for building up vast databases on transcriptional and posttranscriptional interactions. Therefore, we will briefly describe how gene regulation takes place in eukaryotic cells and then introduce methods commonly used to obtain information about gene regulation from biological organisms. We will then review a publication describing concisely, how such data can be integrated in a computational framework, to yield a systems-level understanding of the organism from which the data is derived.

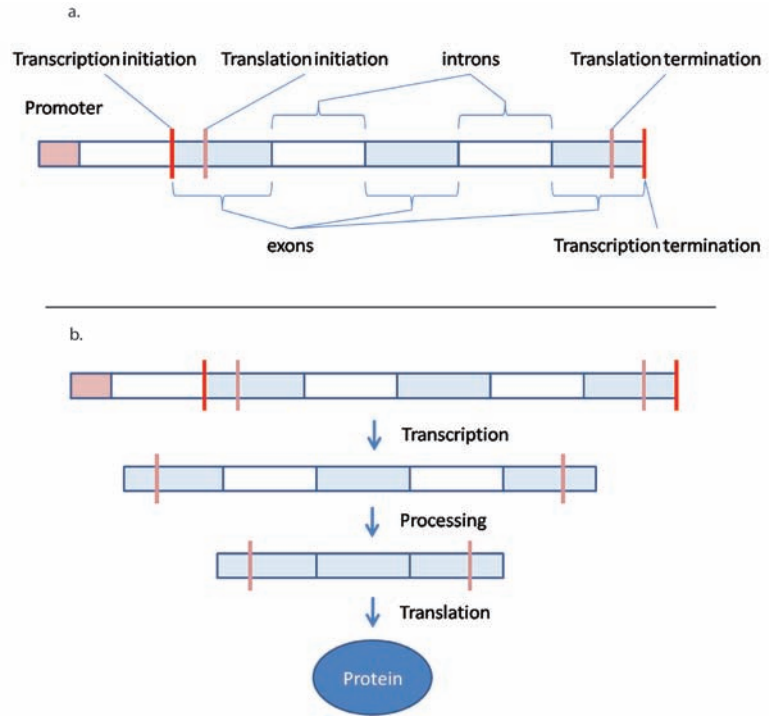
Molecular Basis of Gene Regulation

This section gives a short outline of the processes considered as the molecular basis of gene regulation. For a more complete description, see Alberts et al. (2002) and the further reading section. The DNA encodes all information necessary for the construction and maintenance of the associated organism. Hence, it can be seen as a blueprint, which encodes a temporal and spatial sequence of events that take place during development. Each cell in a multi-cellular organism contains a copy of the same DNA, such that the information is present and can be processed in a parallel manner by each cell separately. The DNA is structured into genes, which in a eukaryotic organism, are typically composed of five distinct functional regions (see Fig. 1a)):

1) a promoter region,
2) the transcription initiation site,
3) the translation initiation site,
4) a sequence of exons and introns,
5) a translation termination site and
6) a transcription termination site.

The process of transferring the information encoded on the DNA to the organism starts with the transcription of genetic information (Fig. 1b). The promoter region is responsible for the initiation of the transcription process. The RNA polymerase, an enzyme which transcribes the genetic information on the DNA into RNA so that it can be further processed, binds to the DNA at the promoter region. The transcription process is then started at the transcription initiation site. The RNA polymerase shifts along the DNA and copies the information into an RNA strand known as nuclear RNA (nRNA). This strand of RNA is then further processed to remove certain regions referred to as introns and re-align the remaining

Figure 1. a) A schematic structure of a eukaryotic gene is depicted (compare Gilbert (2003)). Transcription and translation initiation sites are denoted by red lines. b) The process of transcribing the DNA, then removing the introns and translating the mRNA into a protein, is outlined.



regions (exons), which is called splicing. The resulting RNA, called messenger RNA (mRNA) is then transported to and translated on the ribosomes. Ribosomes are complexes of RNA and proteins, which specifically detect nucleotide triplets in the mRNA and accordingly align a sequence of amino acids, which then arrange into a three dimensional structure (folding) thus forming a protein. This protein serves as a basic building block for the cell. E.g., it can become part of the cell membrane or form complexes with other proteins. It can also serve as a signal, which means that it gets involved in the transcription process of genes as explained as follows.

The RNA polymerase needs special proteins to be able to bind to the promoter region of a gene to begin transcription. These proteins are called transcription factors (TFs). There are specific DNA sites where they can bind to at the promoter. These sites can be distinguished into enhancer sites and silencer sites. Enhancer sites attract those TFs, which ease the binding of RNA polymerase to the DNA. If such a TF is present, the transcription process can be initiated. A silencer can be seen as an inverse enhancer: If the TF is present, the probability of transcription is reduced.

The TFs that bind to enhancers or silencers are usually gene products themselves. Therefore, a regulatory link can exist between genes: the product of one gene enhances or silences the activity of another gene, which in turn can regulate the activity of another gene and so on. The resulting interaction network can be depicted as a graph, where genes are represented as nodes and TF-promoter interactions are depicted as the edges. The graph naturally consists of two different types of unidirectional link: activation and

inhibition, depending on the TF and its binding site. We are interested in the resulting gene regulatory networks because they are thought to be responsible for the different complexities of organisms; complexity seems to depend not only on the number of genes that an organism possesses but more crucially perhaps, on how those very genes should interact with each other.

The process of transcription and translation, which includes both biochemical and biophysical processes, is very complex and poses a big problem for computer simulations. We might ask, what level of detail does our model require in order that the essential properties of the modeled system can be plausibly captured? For example, is it important to model transcription and translation separately when looking at coordination of cellular processes such as cell cycle and division, or can we just build a model that has a direct DNA sequence-to-protein mapping? These questions are not trivial and thus need to be kept in mind when interpreting the results of the respective simulation. Of course, the level of abstraction depends on the scientific questions that are to be addressed.

Obtaining GRN Data from Biologic Organisms

A variety of methods exist for investigating the processes involved in gene regulation and for collecting data on the GRNs of living organisms. We will introduce two popular methods that are typically employed. The first method is used to indicate the activity level of genes in a cell during a certain cellular condition, while the second method is used to discover binding events of TFs to particular regions on the DNA. For a thorough description of analysis methods, see the dedicated further reading section and Gilbert (2003).

The microarray technique allows for the monitoring of gene activity (level of mRNA) of many different genes in cells under different conditions. A DNA microarray is a collection of small DNA spots attached to a solid surface (e.g. glass). Each spot has many copies of identical molecules representing one gene. They are either genomic in which they are part of the respective gene taken from a biological individual, or they are short stretches of nucleotides corresponding to part of the gene. The gene has to be identified beforehand and the corresponding molecule must be put in place, such that complementary RNA or complementary DNA derived from experiments can be added for hybridization (selectively connecting complementary sequences). Three steps are performed during microarray analysis. First and foremost, RNA from a cell in a certain condition is extracted. This RNA represents the activity states of the genes, since mRNA strands that result from transcription are only synthesized if a gene is active. The second step consists of reverse transcription from RNA into complementary DNA (cDNA), where nucleotides are labeled with a fluorescent dye. Then, the cDNA is put on the microarray and left to hybridize with the prepared DNA on the DNA spots. The cDNA will only hybridize on those spots containing its complementary sequence. The amount of hybridized cDNA will be proportional to the number of RNA molecules which were initially present in the cell. Finally, a laser is used to excite the dye until it fluoresces, the intensity of which corresponds to the original activity of the respective gene in the original cell. The microarray technique allows for a surveillance and comparison of a large amount of genomic data, which yields both gene activity information under different conditions and information about the regulatory influence of proteins on a large genetic scale.

Chromatin immunoprecipitation (ChIP) is a method to monitor the actual binding of TFs to the DNA *in vivo*. Proteins that are bound to the DNA, which is necessary if they are to act as TFs, can be chemically immobilized and fixed to the DNA (cross-linked). This makes it possible to keep the association between protein and DNA sequence intact for readout. The readout is performed by breaking the DNA

into small fragments while keeping the proteins attached. The types of protein attached to the DNA fragments can then be isolated and analyzed separately. The DNA fraction that is still bound to these proteins can be read out and the respective sequence can be traced to the genetic code of the organism. In this way, the binding site of a protein can be found. A combination of this method with the microarray technology yields a powerful tool for detecting TF-gene-interaction.

Babu et al. (2004) give a review of different analyses performed in biology research -- using data similar to that described above -- in order that the structure of GRNs in simple organisms can be inferred. The knowledge that can be extracted from biological data with respect to dynamical properties and the evolution of such networks is addressed. In the presented approaches, both local and global structural characteristics of GRNs are analyzed. The article gives a good overview about the possibilities and limitations of the methods that are applied in biological data analysis.

Analysis of Regulatory Network Dynamics from Biological Data

The analysis of Nicholas M. Luscombe et al. (2004) shows nicely the importance of investigating the dynamical changes of links in GRNs during the lifetime of an individual when identifying their role in the structuring and maintenance of the organism. Also, their results show a clear necessity in critically reviewing the information that is obtained by different types of analysis of biological data, on the basis of knowledge about conditions and methods applied during data collection.

Luscombe et al. begin their analysis by assembling data for a static representation of known regulatory interactions in *Saccharomyces cerevisiae* (yeast). This data is available from different publications; essentially, it describes 7074 regulatory interactions between 142 TFs and 3420 target genes. The dynamic perspective is achieved by separating the gene expression data into five different conditions that typically occur during the lifetime of yeast: cell cycle, sporulation (production of spores), diauxic shift (a shift between growth phases), DNA damage and stress response. This separation leads to a dynamic view of a biological network on a genomic scale, since these five conditions are a temporal separation of cellular conditions.

The analysis of the five resulting networks shows that the different conditions all affect the network properties on both local and global scales. Firstly, half of the target genes are only expressed in one of the five conditions mentioned above. A closer inspection of the networks leads to a separation of characteristic networks which can then be linked to two different kinds of condition. The authors refer to these conditions as endogenous processes, which reoccur periodically as part of ‘normal’ cell life (e.g. cell cycle), and exogenous processes. These processes are binary events triggered by external factors such as DNA damage or stress response. Topological measures applied to these different networks change considerably between the two stages. The authors analyze in-degree, out-degree, path length and clustering of the networks, and conclude that depending on the type of condition the cell is in, an optimal strategy is chosen to either react quickly (exogenous conditions), or robustly (endogenous conditions). These two strategies necessarily show different network topologies. The second analysis deals with motifs, which are local features that further serve to characterize network behavior. Again, the relative number of characteristic motifs, such as the single-input motif, the multiple-input motif and the feed-forward loop motif, change considerably under different conditions.

Statistical analysis yields information on the presence of hubs across the conditions. A regulatory hub is a TF that regulates a dis-proportionally large number of target genes. The authors find that most

hubs (78%) are transient across conditions, which means that they are only strongly influential in one condition, and less so in the other conditions.

The implications of this work to systems biology research are twofold. Firstly, the structure and function of GRNs of biological organisms both change during the lifetime of an individual. Therefore, in order to gain insight into biological systems on a holistic level, dynamical models for the representation of such systems and the methods for their analysis need to be created that can incorporate this important aspect. Secondly, independent of whether this data serves as the basis for the design of a technical application, or whether a computational model is to be designed for additional research, (e.g. evolutionary studies from a systems point of view), there is a clear necessity to understand and assess the fundamental conditions and assumptions, from which the biological data is derived, for example, the condition a cell is in, when GRN data is collected.

Computer Simulation Tools in Systems Biology

The use of computers gains importance and leads to new insights in biological research. As seen above, it is common to use computational tools for the statistical analysis of network data, gathered from biological experiment. Another use of computers in systems biology is to replace the biological experiment in part or completely by simulation. The reasons for choosing a computer simulation instead of a wet lab equivalent is manifold: biological experiments are often complex and expensive, or they are sometimes very sensitive to contamination such that only a small percentage of experiments actually yield any results at all. Also, many processes *in vivo* have their own timescale, which often gives rise to a problem of observation (both cases: if the process is too fast, observation is problematic due to the time resolution of the analysis tools, or if it is too slow, it is often not feasible to study processes over several generations of a species, for example). Another crucial point is the general ability to completely monitor a complex living system: measuring all processes and influences that might be the reason for certain observations is often a practical impossibility. Problematically, *in vivo* experiments often only investigate a small fraction of cellular process while other cellular activities, which would ideally be isolated and separated in order to remove their influence, cannot be easily stopped.

By contrast, computational models are cheap and clean, which usually means that they do not need expensive hardware and are not distorted by extraneous influences arising in the laboratory. Usually, a computational model can be chosen having a trade-off between speed and accuracy, which makes experimental research fast, and at the same time provides the ability to critically assess the results regarding significance. Another very important point is the possibility to store the complete data set from an experiment so it can be accessed multiple times for analysis. Thus, experiments are traceable, and repeatable in exactly the same way. Also, the well defined number of interacting components allows for a limit in complexity of the experiments, which eases any subsequent analyses.

The major problem for such computational models lies in the choice of the level of abstraction. This choice necessarily neglects some parts of the biological system, which in the worst case might be a component that strongly influences the process under investigation. Hence, a critical assessment of simulation results must be performed: firstly, by statistical analysis and secondly, by observation and validation in vivo.

Computational method in systems biology leads to a new and exciting opportunity for research: the observation of evolutionary process becomes tractable. Simulated evolution, based on simulated

mutation and selection yields a powerful tool to investigate the reasons for the formation of different features in biological systems, such as the evolutionary emergence of motifs (e.g. feed forward loops) in GRNs and as an evolutionary basis, it gives credence to the discussion about the omnipresence of so called scale-free topology in biological networks (Keller (2005)). These simulations can either take as a starting point for a repeated evolutionary cycle of mutation and selection, a model of the GRN of a real organism (e.g. derived from bacterial DNA), or start with a completely random initialization.

In the following section, we will review a few interesting approaches to evolving artificial GRNs from scratch to produce biologically motivated signaling behaviors. Then we will present a model that we developed for the investigation of evolving stable growth in multi-cellular organisms.

COMPUTATIONAL MODELS FOR THE EVOLUTION OF GRNS

Introduction

Having presented some of the computational methods for analyzing GRNs that are employed in biology research, we now turn to a more abstract use of computer simulations for the integration of an evolutionary perspective into GRN analysis. In doing so, we will review a few works which are all similar in approach but in contrast to the work described in the last section, they use computational models for the generation of experimental data. We briefly introduce works of interest for reference, and then concentrate on two articles that show nicely the evolution of artificial GRNs to carry out specified functions. These works are of great interest to researchers both in biology and in computer science, since on the one hand, they explore the space of possible solutions of GRNs that are evolvable *in silico*, and on the other, they address basic biological questions such as robustness of GRN dynamics and functional properties of GRN motifs.

It must be kept in mind however, that the simulated evolution in these cases usually takes inspiration from the evolutionary computation paradigm (Fogel (1995)) rather than from a biological, population genetics point of view. Accordingly, an abstract view of encoded information, where mutation is represented by an uncorrelated random change, is often preferred over a more biological perspective; in such view, there is no consideration of environmental constraints, genetic drift and linkage influence on the evolution of genotypes (Lynch (2007)). To account for such effects in computer models of GRN evolution, and to investigate the resulting features, for example, allelic diversity in populations, a more detailed simulation of environmental diversity, genetic representation and mating behavior would be necessary.

Designing GRNs with Specified Function

Different approaches exist to model GRNs which exhibit a particular functionality. Chen & Wang (2006) pursue an approach based on mathematical modeling of GRNs. Their model is based on *a priori* knowledge about the structure of functional modules such as bistable switches and oscillators. They make a theoretical analysis of the presented networks to determine the dynamic behavior and thus give precise guidelines for the artificial creation of GRNs with a desired behavior.

Evolutionary methods for the design of GRNs are either employed in artificial life experiments or in biology research. An artificial life approach is given by e.g. Rudge & Geard (2005), where a slightly

modified artificial neural network replaces the usual GRN model for the control of multicellular growth. It is shown that the network can evolve towards controlling the growth of leaf-like structures.

In the following section, we will concentrate on two representative biologically motivated models that show nicely, the use of computational methods to evolve GRNs with a specified function.

Evolving Switches and Oscillators *in silico*

An interesting application of simulated evolution of GRNs is presented by Francois & Hakim (2004). Francois and Hakim investigate the evolution of small gene networks which have the ability to perform a very basic task. They use simulation *in silico* without the introduction of *a priori* knowledge about the topology of these networks. The goal of this research is to understand the structure and dynamics of small functional building blocks in GRNs, since the investigation of the function of a given motif, as well as the finding of a suitable motif which should perform a given function, is not as straightforward as it seems. Usually, only a part of the interactions in a regulatory motif are known, and to realize a given function, a multitude of motifs could be used, sometimes even depending on the rate equations used for the computational model.

Francois and Hakim use the simulated evolutionary process to determine possible motifs for a given function. They choose two target functions, namely an oscillator and a bistable switch, which they select for, during evolution. A bistable switch is a small network motif consisting of at least two components, which has two different attractor states. Depending on the initial condition, the system converges into one of the two states.

The simulated genetic networks in the model of Francois and Hakim are defined by a number of genes, proteins and deterministic rate equations. Interactions take place in the form of activation or repression of gene expression through proteins and posttranscriptional interactions between proteins. The simulation starts with two genes and randomly drawn production and degradation rates. Evolution advances by creating a population through mutation, evaluation and selection. Mutations can take place in the form of a change of production or degradation rate, a change of reactions and their kinetic constants, a creation of a new gene, creation of a new interaction between protein and gene promoter and by addition of a posttranscriptional reaction.

The results presented by Francois and Hakim show that both switches and oscillators are evolvable. Furthermore, the evolved GRNs are able to perform for a wide range of input parameters, and they exhibit a good performance in the presence of noise. The authors also identify motifs in biological GRNs that resemble those found in their *in silico* evolutionary process: Bistable switches with a comparable architecture are found in the *lac*-operon, and in Xenopus oocyte maturation under special conditions. Another evolved bistable switch can be compared to a motif found during *B. subtilis* development. One of the oscillators which evolved in the experiments can be matched with the architecture of a biological circadian clock.

However, evolving sustained oscillatory dynamics based on computational models of gene regulatory networks has been found to be nontrivial. To address this problem, various fitness functions to facilitate the evolution of oscillatory dynamics have been suggested, see e.g., Paladugu et al. (2006) and Chu (2007). Our recent work (Jin & Sendhoff 2008) has also disclosed that it is quite difficult to evolve stable oscillation for gene regulatory models using differential equations based on a Hill function. However, it becomes much easier if a step function is used.

In general, the above work sheds some light on how the problem of designing networks with a desired behavior can be tackled, and how it is possible to relate *in silico* evolved networks to biology. If such comparisons are biologically sensible, it might also be useful for answering questions about how such motifs evolved in biology, and especially to find evolutionary advantages in comparison with other realizations of similar functions that are not seen in biology, i.e. which occur during evolution and at least partly exhibit the desired behavior, but still are not persistent throughout further generations.

Evolving Circadian Clocks

Here, we review an application of a computational method to the investigation of a typical biological phenomenon, namely circadian clocks. Circadian clocks are biochemical processes inside cells that emit periodical behavior and are tuned to the terrestrial dark-light cycle of day and night. Knabe et al. (2008) study the simulated evolution of artificial GRNs with the evolutionary goal to realize simple models of biological clocks which respond to periodic environmental stimuli. The investigation is based on simulations which consider different external conditions in the evolutionary process. The influences of a noisy periodical input as well as blackout periods, where the external signal is not present for a certain amount of time, are investigated. Also, different types of external stimuli are simulated, such as a sine curve input signal and a pulse shaped input signal. The approach is used to address questions about the necessity of external signals for such periodic processes in nature, as well as questions about the evolution of the ability to adapt to perturbations in cycle length, phase shift and resetting of the external signal.

The experiments performed are *in silico* simulations of evolutions of single cell organisms. An artificial DNA, which encodes genes with TFs as gene products as well as cis- regulatory modules that bind TFs as enhancers or silencers, forms the basis for the evolutionary runs. The input signal is presented as a periodic modulation of one TF, while another TF is seen as the output of the system.

For the experiments, an initialization of the artificial DNA with random values is performed and a standard genetic algorithm with simulated mutation and crossover is used. The simulated evolution then yields artificial GRNs, which have the ability to produce periodic behavior on the basis of a periodic input, independent of the type of input (sine/pulse). Results are e.g. the possibility to evolve periodic behavior with and without phase shift between input and output, and internalization of the periodic rhythm such that in a blackout phase of the input, the output continues with periodic behavior. Interestingly, the internalized rhythm has a slightly smaller period than the external one, in the case when the external signal is switched off. This phenomenon is similar to many biological organisms, which turn to a slightly different rhythm of day and night activity if an external daylight stimulus is not present. Another interesting finding is that evolved solutions are able to cope with a phase resetting in the external signal, even if this condition has not been encountered during the evolution of the GRNs. Therefore, coping with phase resetting seems to be an inherent ability of these successful circadian clocks. However, in which way and how strong the evolved GRNs rely on input from the environment turns out to be strongly dependent on the conditions under which they evolved, e.g. the presence of noise or a change of input signal from sine wave to pulse.

This kind of research shows nicely, how the simulated evolution *in silico* yields new insights into biological problems, and now makes it possible to hypothesize mechanisms underlying the features of biological circadian clocks and also to specifically validate these results *in vivo*. The simple GRN model allows for fast computation and easy observations of the dynamic processes, as well as the evolutionary steps which lead to them. Manual changes, such as switching between sine and pulse input is easily

achieved, and easy repetition of the experiments makes it possible to perform statistical analysis. The insight into evolutionary history, which can only be gained by such computational experiments, is a new perspective that could greatly facilitate GRN analysis in biology in future.

EVOLVING STABLE MULTI-CELLULAR DEVELOPMENT

Introduction

We now present our work on modeling and evolving GRNs *in silico*. It is inspired by an evolutionary development model presented by Eggenberger (1997). Modeling evolutionary development has recently been of growing interest to the engineering and computer scientist community, since it is regarded as a possible approach to the design of complex systems. Related approaches can e.g. be found in Bowers (2006), Bentley and Kumar (1999), Federici (2004) and Miller (2003). We want to use simulated GRNs to control multi-cellular growth, since multicellularity poses many unanswered questions, e.g. how is the timing of events in a multicellular organism coordinated? How is cellular differentiation accomplished? Which mechanisms underlie stable development and finite growth? In the following, we will investigate the possibility to model a limited cell growth, which means that a finite number of cells is produced and no further cellular division occurs within a limited period of time. This limited growth is achieved when the dynamic system of the gene regulatory network reaches a stable attractor, which is a nontrivial task in simulated multi-cellular development.

The possibility to investigate dynamic and static features of simulated GRNs enables us to perform an analysis, which in an *in vivo* experiment would be impossible: we screen a number of individuals and their offspring throughout the evolutionary process by recording the genetic interactions in every developmental timestep. These interactions represent the complete dynamic behavior of the GRNs. We search for dynamic negative feedbacks, and investigate their role in mutational robustness. Another very recent work by Kwon and Cho (2008) investigates the link between feedback and robustness. However, the results do not take the evolution of networks into account. Interestingly, the role of negative feedback without the presence of evolution seems to be the exact opposite as described in the following.

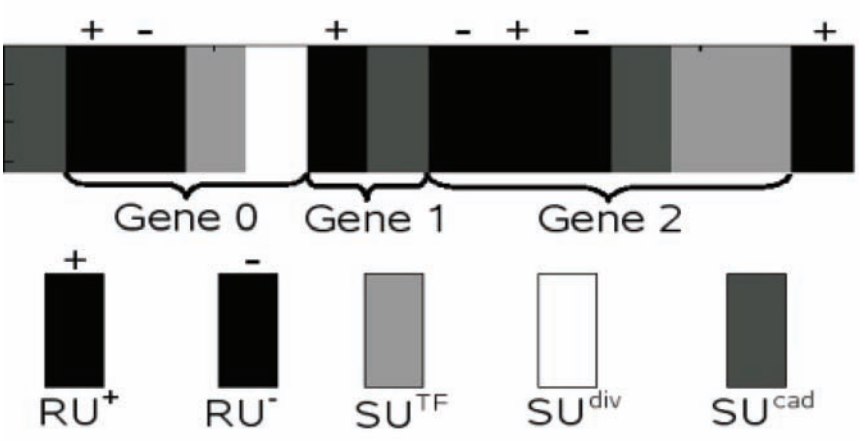
We organize the section as follows. First, we describe the used genetic representation in detail and continue with a description of the cellular model for multi-cellular growth. Then we will discuss the experimental setup together with our visualization and analysis methods. Finally we discuss experimental results which elucidate the role of an emerging feature in the evolved GRNs, namely negative feedback.

Gene Regulatory Model for Cell Growth

The model overview in this chapter is kept concise to allow for a focus on description and analysis of simulation results and their implications. For a more thorough model description, please refer to Steiner et al. (2006; 2007).

In the model, cellular growth is controlled by a genome stored inside a virtual DNA (vDNA), a copy of which is available for translation in all cells of an individual. This genome consists of regulatory subunits (RUs) and structural subunits (SUs), which are initially lined up randomly. A functional unit of this DNA, called a gene, is composed of a group of SUs and the preceding RUs. The SUs encode

Figure 2. An illustrative vDNA with three genes, each consisting of one or more structural subunits (SUs) and regulatory subunits (RUs). Two different kinds of RUs exist: silencer RU^- and enhancer RU^+. A SU coding for the production of a transcription factor (TF) is denoted by SU^{TF}, a SU coding for a division by SU^{div} and a Cadherin producing SU by SU^{cad}.



actions that a cell should perform, while the RUs determine whether a functional gene is active or not. The actions encoded for in the gene will be performed only if it is active.

An illustrative example of a genome with three genes is given in Figure 2. Note that the RUs behind the last SU and the SUs in front of the first RU are ignored during the developmental process. SUs and RUs as building blocks of the vDNA are both represented as vectors of double precision values. The following description clarifies the setup:

- Structural subunits: A SU encodes the action to be performed by a cell, and contains the parameters that specify the action. Possible actions include cell division, production of a diffusing chemical, and, the transcription factor (TF) for cell-cell signaling, and production of Cadherin molecules on the cell surface, which determine cell-cell adhesion forces. For division, only one value inside the SU is used, which codes for a division angle. A TF is encoded by four parameters: a label, a production rate, a decay rate and a diffusion rate. Cadherin molecules are encoded by one parameter which represents a type. Adhesion force calculation is then based on the similarity between Cadherin types on different cell surfaces.
- Regulatory subunits: Two types of RUs are used in our model, either activating (enhancer) or repressing (silencer) the expression of a gene. RUs can sense the presence of certain types of TFs in the vicinity of the cell. If the label of a TF is affine, i.e. numerically close to an associated label within the RU, and if the concentration of the TF lies above a threshold, which is also encoded in the RU, an activity value is determined for each RU. All activating (= positive sign) and repressive (= negative sign) activity values belonging to the same gene are used to determine the overall activity of the gene.

Cells and Their Interactions

We define the simulation area for cellular growth by an equally spaced 26 by 26 grid with a step size of 0.5, on which the concentrations of the TFs are calculated. Cells are modeled as spheres with a radius of one, and are not allocated on the grid, contrary to the TFs. They interact with each other by reading TFs from and releasing TFs to the nearest grid points and by cellular motion through rigid body interactions coupled with adhesion forces. For details of the implementation and the mathematical formulation of cellular interaction, please refer to Steiner et al. (2007).

The ability of TFs to diffuse is inspired by the biological way in which cellular communication is achieved: The release of signaling molecules by one cell and their diffusion to neighboring cells, where they trigger a signaling cascade to transfer the signal to the inside of the cell and thereby regulate gene expression, results in so called positional information. This information from the neighborhood causes different genes to be expressed in different cellular contexts. Note that for simplicity, instead of simulating signaling cascades, we use the concentration of a TF at a cellular position directly as 'input' to the RUs of the genes.

Time Scales and Sequence of Events

In the beginning of development, a single cell containing the vDNA is placed at the center of the simulation area. To start the growth process, an initial TF (maternal TF) is released, which maintains a constant concentration in the whole area over the entire developmental time. The initial TF concentration in our model does not provide any positional information. Rather, it fulfills the minimal requirement for starting a developmental process since it can be seen as a constant environmental influence (e.g. in contrast to a periodically changing environment).

In each developmental step, the following events take place. Firstly, the translation of the DNA is initialized for all existing cells. Secondly, if the TFs in the vicinity of the cell are affine to a RU and exceed the threshold encoded in the RU, they activate the gene, and the action that the gene encodes is executed. If a division gene is active, a new cell is placed inside the calculation area, close to its mother cell with an overlap. Finally, the position of all cells is updated to minimize overlap, and the diffusion simulation of the released chemicals is advanced in time. The whole process repeats until a termination criterion is met, i.e. until a stable state is reached or a maximal number of developmental time steps have passed.

The Selection

Having introduced our model, we will now describe the way we use this *in silico* simulation for the investigation of evolving GRNs.

From earlier investigations, we know that most randomly generated individuals cannot produce limited growth: either the initial cell does not divide at all, or new cells spread over the whole calculation domain in a cancer-like growth. Therefore the goal of evolution in our experiments is to find GRNs that result in a stable developmental process, which means that a stable state must be reached before a maximum number of developmental time steps is reached.

A development is stable, if it reaches a state where cells no longer move or divide, which can be identified by the observation that the concentration of the TFs either have decayed to a value below

all activation thresholds, or have reached stable values, which indicates that no further change in gene activity can occur.

To have a function where fitness can increase continuously such that evolution can progress, the finite number of cells that make up the individual should be located inside a predefined diamond shape centered in the calculation area.

The evolution of limited cell growth is formulated as a minimization problem. The fitness *f* is defined by the following equation:

$$\cdot_i = \begin{cases} -1 & if \quad \left\| p_i \right\|_1 \leq 5 \\ +1 & if \quad \left\| p_i \right\|_1 > 5 \end{cases}$$

$$f = \sum_{i=1}^{N} \cdot_i ,$$

where p_i is a two-element vector containing the position of the *i*-th cell of the individual in the last time-step, *N* is the total number of cells, and $\|.\|_1$ denotes the 1-norm. In other words, the fitness is expressed by the number of cells outside a diamond shape around the center of the calculation area, deduced by the number of the cells inside the diamond shape. Constraints ensure that only individuals which produce stable growth are selected: Firstly, if the cells touch the border of the simulation area, the growth is presumably uncontrolled and infinite. Secondly, if the growth process does not reach a stable state within a maximum number of developmental time steps, it is not suitable for our purpose. In both cases, a penalty term of +700 is added to the fitness function, which is high enough to ensure that these individuals are discarded during the selection process. Note that in this setup, the smaller the fitness value, the "fitter" the individual is.

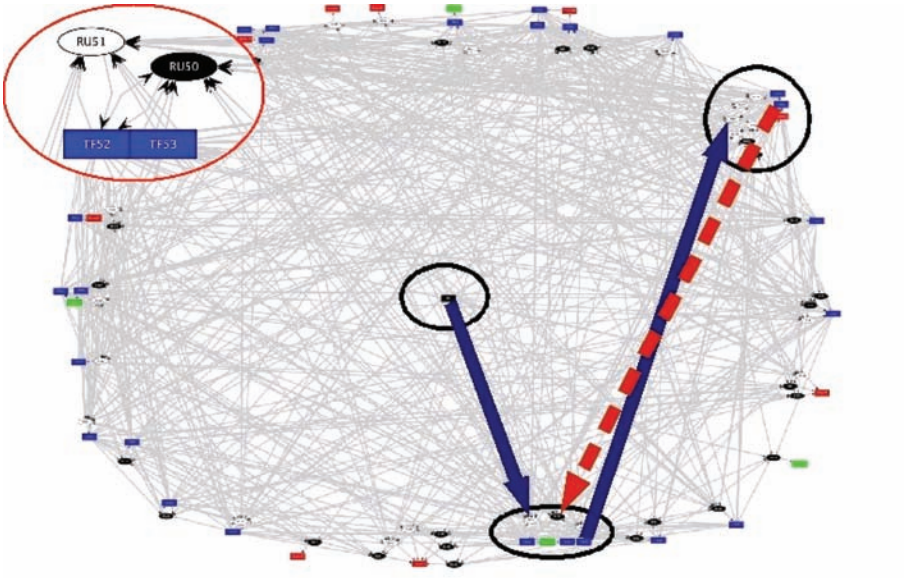
Static Links in Simulated GRNs

We depict the static interactions of a GRN in Fig. 3. It belongs to a successful individual of an evolutionary run described in the next section. The static interactions can be directly derived from the vDNA of an individual in the following way: The type of a TF is encoded in the SU of a respective gene. At the same time, each RU has an affinity parameter, which is compared to the type of a TF if present, to determine whether the TF can bind to the RU at all, independent of its concentration. This information leads to the static GRN: if the type of a TF encoded in a SU is close to the affinity parameter encoded in a RU, a regulatory link is drawn from the SU to the RU.

This kind of representation can be useful for an overview over possible interactions, although the generally high number of interactions makes it hard to analyze them in detail. In general, this representation resembles the genome wide static interaction map which is commonly found in biology for different organisms such as yeast.

The major drawback of this visualization method is that it does not become clear, which interactions really become activated during development. The reason is that the real interaction between a TF-coding gene and a RU depends on thresholds and the concentration of the TF. The concentration then depends on the position of the cell in which the gene is active, the expression rate of the TF in all surrounding cells and thereby on the actual developmental time step. Therefore, to gain an insight into the real interactions, the missing information - TF-concentrations and time steps - needs to be included.

Figure 3. The static interaction network of an individual from the evolutionary run. The pre-diffused TF is placed at the center of the calculation area. A close-up on one gene is depicted on the upper left side of the figure: the gene consists of a silencer RU (black ellipses), an enhancer RU (white ellipses) and two TF-coding SUs (blue rectangles). Two interacting genes and the pre-diffused TF are emphasized by bold circles. A solid arrow from the pre-diffused TF to the lower gene denotes an activating connection, which could be the starting point of a negative feedback loop between the two marked genes (the dashed arrow denotes a repressing interaction). Note, however, that the analysis of the dynamic GRN reveals that this feedback is not used, because the concentrations of the TFs do not exceed the threshold values.



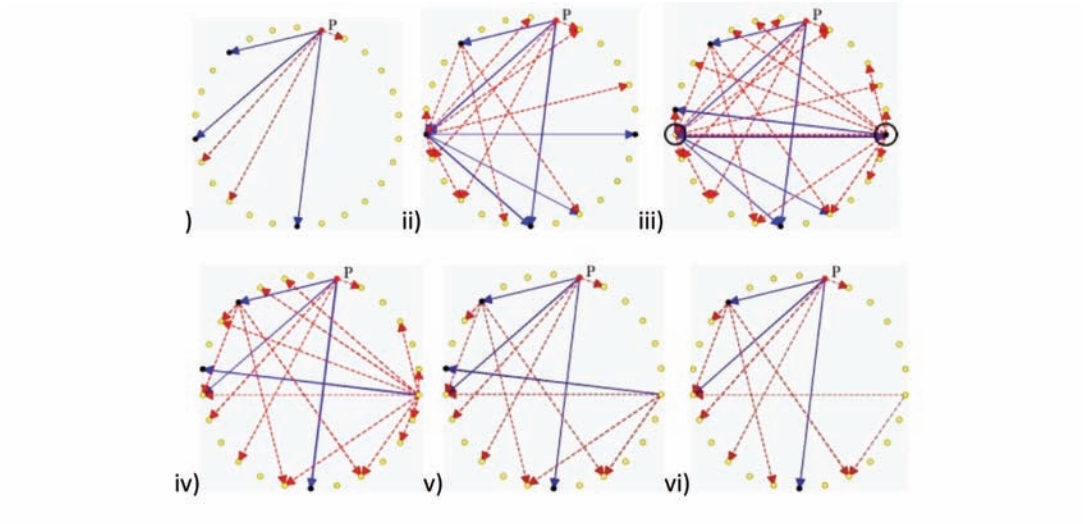
Dynamic Links in Simulated GRNs

In Figure 4, we depict a time series of network interactions as they take place in the first cell of an individual. Genes are represented by points and arranged in a circle. Since information about TF concentrations in the vicinity of the cell can be obtained for every developmental time-step in the computational model, the real interactions between genes can be shown here. At each time step, the interactions are updated according to the changing TF concentrations. The top solid point in Fig. 4i), marked by 'P', denotes the pre-diffused TF and therefore, exhibits initial interactions. From there, gene activation and inhibition can be tracked in each successive time step, from Fig. 4 i) to Fig. 4 vi).

Note that this dynamic representation of the GRN can differ from cell to cell. For our experiments, we checked that all cells of one individual reach the point where the GRN converged to the same stable state. Therefore, our analysis is performed only for the first cell of an individual.

We use the information provided by the dynamic GRN for negative feedback analysis. In every developmental time step, we search for closed loops in the GRN and count the number of negative interactions which are part of the loop. This is achieved by transforming the network into a tree-graph and looking for the occurrence of already visited nodes by stepping along the tree. The method yields the number of negative feedback loops for all developmental time steps in one individual. By comparison,

Figure 4. A time series of interactions inside the dynamic GRN. Each gene is depicted as a small circle. The red point denotes the pre-diffused TF. Active genes are marked as filled circles. The interactions between the genes are either repressive (red, dashed arrows) or activating (blue, solid arrows). In iii) we highlighted two genes that form a negative feedback loop with an activating interaction from left to right and a repressive interaction in the opposite direction. Each figure represents the state of the GRN in one time step. Note that the static condition for this individual is not yet reached after time-step vi).



we can eliminate the occurrence of the same loop in successive time steps and thus find the number of unique negative feedbacks used throughout the developmental process. The following analysis can now be based on these dynamic interactions, which really occur during simulated development. In the review of Luscombe et al. (2004) above, we have already seen that the dynamic GRNs differ greatly from the static ones even in biology. Therefore, such an analysis is necessary and may yield new insights into developmental processes in general.

Evolution of Stable Growth and Emergence of Feedback

Simulated Evolution

We use a standard evolution strategy (ES), which is commonly used for computational optimization of engineering problems, as described in e.g. Schwefel (1995). ES is based on vectors of real valued parameters, which are called chromosomes. In our case, the vDNA is such a chromosome. These chromosomes encode the process or designs that are to be optimized. Each chromosome belongs to a so called individual. This individual represents an individual solution to the given problem. A population of individuals is formed by simply gathering a certain number of these individuals.

On this basis, an evolutionary cycle can be described as follows: After an initialization step, where a population of individuals with randomly initialized chromosomes is generated, the following three steps

1. Evaluation
2. Selection
3. Mutation

are repeatedly executed. In the evaluation phase, a decoding from the data in the chromosomes into the respective designs is performed. This corresponds to the simulated growth process described in our experiment. Then a fitness function is used to determine the quality of the resulting design. During the selection phase, a fraction of the individuals in this population is selected for reproduction, while the remaining individuals are discarded. Finally, mutating the selected individuals several times yields a new population for evaluation. This process is stopped after a certain number of repetitions (generations) or if some individual reaches a desired quality.

External Conditions for Simulated Development

Since external factors, such as pre-diffused TFs or the choice of maximal growth time affect the growth process, we will shortly provide information about the way we choose these boundary conditions for the simulated development.

In biology, it is often possible to observe the presence of predefined gradients of maternal TFs in the egg cell, facilitating polarization and axis formation. In our case, we do not provide such a gradient since it is important to us that the finite number of cell divisions is a result of a GRN reaching a stable state, and not a mere simple interpretation of external cues. Therefore, the concentration of the initial TF is equally distributed across the diffusion grid. Every grid point possesses the same amount at all times during growth. Thus, neither positional information, nor temporal differences can be used by the growth process to easily achieve finite growth.

The maximal developmental time is arbitrarily set to 100 developmental time steps, i.e. 100 rounds of processing of the vDNA. Of course, each individual that reaches this limit without stabilizing growth is not feasible and thus cannot reproduce.

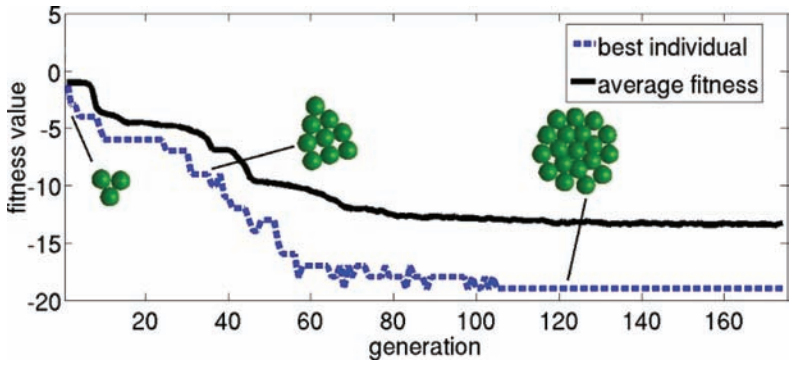
We will now proceed with a description and discussion of the experimental results using analysis of dynamic GRNs.

Results

Stable Growth

The result of a typical evolutionary run is presented in Fig. 5, where the fitness of the best individual and the average fitness are plotted. It can be seen from the figure that the population stagnates from time to time, before an innovation is found, which leads to a significant fitness increase. A much wider plateau has also been observed in some of the runs. Note that the goal of the experiment is not to show how well the diamond shape can be realized. Instead, our model serves to analyze systematic features of the simulated process. Successful individuals exhibit the non-trivial behavior to grow towards a stable state during their development. This means that their shape and final state of the GRN remain constant after a certain developmental time step. The dynamic GRN representation can now be used as a tool for analyzing different features in the evolution of developmental control resulting in this stable growth. We look at an interesting dynamic motif, namely negative feedback, as it evolves over generations. From

Figure 5. The best (dashed line) and average (solid line) fitness of a typical evolutionary run. The shape of the best individual after convergence to a stable state is shown for three different generations. The average fitness is computed only from those individuals that do not violate constraints.

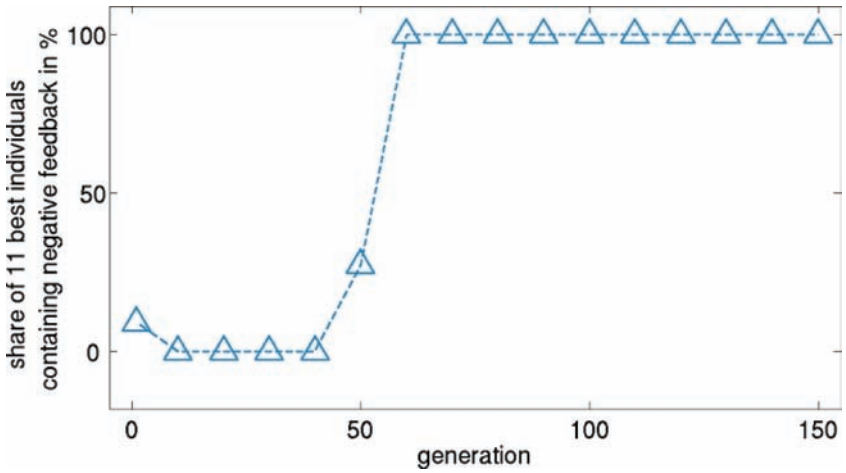


control theory, we know that negative feedback might act as a stabilizing component in a linear dynamic system, though it is not always the case in a nonlinear system (see Nicolis, 1995).

Feedback

The curve in Figure 6 shows the emergence of feedback during the evolutionary run. Since the analysis is computationally expensive, we chose to test the 11 best individuals at every ten generations for feedback. The curve shows clearly, that negative feedback starts to prevail between the 40th and 60th generation. After generation 60, all 11 best individuals contain feedback loops. We are able to track the first occurrence of feedback back to the best individual of generation 44, whose dynamic GRN is depicted in Fig. 4. The negative feedback is visible in Fig. 4 iii): an activating connection from the highlighted gene on the left side to the highlighted gene on the right side, and a repressive connection in the opposite direction.

Figure 6. The triangles mark the percentage of the 11 best individuals which possess one or more negative feedback loops. The analysis is performed at every 10th generation.

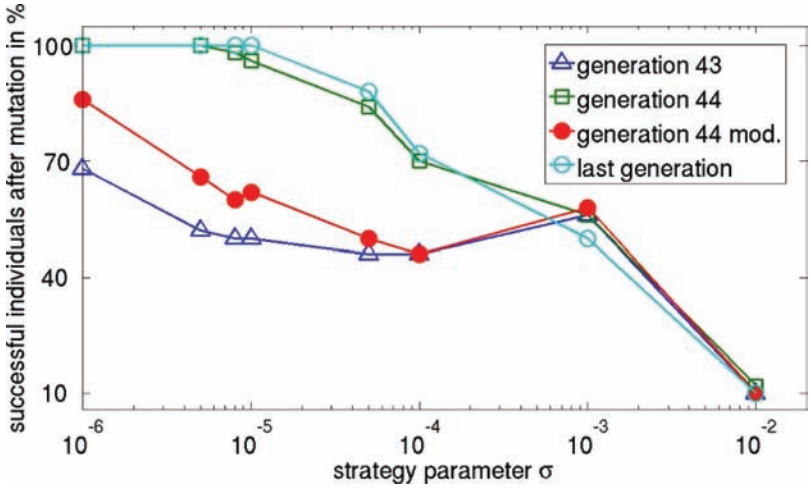


We assume that negative feedback stabilizes the development of individuals against mutation. If the concentration increases beyond a defined threshold, the TF can decrease its own production. If the concentration decreases, the level of self influence is reduced resulting in a stable state. In comparison, a positive feedback loop could only cause a TF to increase its own production continuously without reaching a stable state. One possible effect is that offspring of individuals with negative feedback will be less sensitive to mutations, i.e., fewer lethal mutations will occur. Here "lethal" means that individuals will not grow at all or will not reach a stable state after the maximum number of allowed developmental time steps. In both cases, individuals are penalized, and not taken into account for further selection. Thus, the number of feasible offspring from an ancestor containing negative feedback loops is higher than the number of feasible offspring from an ancestor without negative feedback. If the fitness of individuals containing feedback is not worse than the fitness of those without feedback, the probability that a genome with feedback is passed on during evolution increases.

To verify this hypothesis, we perform a simple mutation experiment with four different individuals: The best individual from generation 44 which uses feedback, its direct ancestor from generation 43 which has no feedback (see the static GRN in Fig. 3), the best individual at the end of the evolutionary run and a modified version of the best individual from generation 44. The modification consists of removing the gene from the vDNA that causes the negative feedback, marked in Fig. 4 iii) with the rightmost circle. Note that these individuals still exhibit a stable, finite growth process, thus none of them violate the constraints.

The four individuals are mutated 50 times each, for every sample point. Mutation is carried out by adding a random number generated from a zero-mean normal distribution with given standard deviation σ to each value of the vDNA. Thereafter, we count the number of individuals that still produce stable growth without violating the constraints and denote them as successful. Note that feasible individuals with a lower fitness than the unmodified ones are also among them. Fig. 7 shows the results of this experiment.

Figure 7. The results of the mutation experiment: four individuals are mutated 50 times for each strategy parameter, i.e. mutational strength σ. The plot shows the percentage of individuals that survived mutation.



It is clearly shown that mutations with σ smaller than 10^{-5} affect individuals without feedback much more severely than individuals containing feedback: 100% and 96% respectively of the individuals containing feedback survive, while only 62% and 50% respectively survive without feedback. At $\sigma = 10^{-4}$, feedback is still an advantage, although the percentage of successful individuals has been reduced significantly to 70%. The percentage of lethal mutations with a σ larger than 10^{-3} is similar for all individuals. This might be the result of mutation destroying the negative feedback loop, thus destroying the whole control mechanism that mainly set the different individuals apart.

CONCLUSION

Simulated evolutionary development is of great interest for two principal reasons. On the one hand, as mentioned above, it provides support for biological research, and on the other hand, technical systems design can profit from such models, e.g. through learning about dynamical features of certain network motifs.

Technical Value of Simulated Evolutionary Development

Biological systems possess many desirable features for technical applications. Robustness towards noisy or faulty inputs, good performance under a multitude of conditions or in changing environments and evolvability are just a few examples. Therefore, it is desirable to understand the mechanisms underlying biological systems which bring about these features. Such mechanisms can be included in the design process of technical systems. For example, one could imagine the simulated growth of a complex inner structure. Instead of encoding all positions of voids and their shapes, a developmental process can be used where material grows to yield a suitable design. This growth could be controlled by just a few variables, providing a significant advantage for optimization processes, as well as for the scalability of such approaches.

Support for Biological Research

We finally want to point out again that the combination of knowledge on evolutionary history and dynamics of developmental processes is important for understanding principles that govern biological systems. Computational approaches are indispensible for achieving such an understanding and for testing certain hypotheses. Combining computational models of evolution of GRNs and systems biology research will give a new insight into biological processes on an evolutionary scale, which then can be used to augment investigations in biology, e.g. dynamic patterns or motifs that play an important role in simulation can be traced in different organisms and their functions understood. In the case of robustness against mutation, the occurrence of feedback-loops in GRNs of organisms that adapt to an environment that causes high mutation rates (e.g. places with high radioactivity) could be investigated and then a comparison with other organisms that evolved in less mutational conditions could be performed for a better understanding of biological adaptation mechanisms.

Future Research Directions

It is clear that a better understanding of the relation between network features and resulting system dynamics is necessary to advance many different fields of applications, such as e.g. molecular medicine, agriculture, and complex systems engineering. In addition to this point, there exists the problem of assessing the *changes* in dynamics, resulting from a network change. Although mutational experiments in the light of (artificial) evolution are a first step in this direction, the resulting changes are far from understood.

Another challenge lies in the integrative nature of complex systems and organisms: Usually, they are part of a complex environment which directly influences their own structure and behavior. We must try to take this environment into account for research to gain a more holistic view of the structure and dynamics of individual organisms, and of complex systems in general.

Finally, the choice of the right level of abstraction for a given scientific question, especially when using computer models, should itself be considered a research question. Having a formal understanding of the effects and restrictions of modeling details on the simulation outcome would allow us to efficiently choose the right level of abstraction and to predict the degree of generalization of results.

REFERENCES

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). *Molecular biology of the cell, fourth edition.* Garland Science, Taylor & Francis Group.

Babu, M. M., Luscombe, N. M., Aravind, L., Gerstein, M., & Teichmann, S. A. (2004). Structure and evolution of transcriptional regulatory networks. *Current Opinion in Structural Biology*, *14*, 283–291. doi:10.1016/j.sbi.2004.05.004

Bentley, P., & Kumar, S. (1999). Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. *Proceedings of the Genetic and Evolutionary Computations Conference 1999*, Orlando, Florida (pp. 35-43).

Beyer, H., & Schwefel, H. (2002). Evolution Strategies - a comprehensive introduction. *Natural Computing*, *1*, 3–52. doi:10.1023/A:1015059928466

Bowers, C. P. (2006). *Simulating evolution with a computational model of embryogeny.* Doctoral Dissertation, The University of Birmingham, UK.

Chen, L., & Wang, R. (2006). Designing gene regulatory networks with specified functions. *IEEE Transactions on Circuits and Systems*, *53*(11), 2444–2450. doi:10.1109/TCSI.2006.883880

Chu, D. (2007). Evolving genetic regulatory networks for systems biology. In *Proceedings of the Congress on Evolutionary Computation 2007*, Singapore (pp. 875-882).

Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. *Proceedings of the 4th European Conference on Artificial Life* (pp. 205-213).

Federici, D. (2004). Using embryonic stages to increase evolvability of development. In J. Miller (Ed.), *Proceedings of the Workshop on Regeneration and Learning in Developmental Systems WORLDS 2004.*

Fogel, D. B. (1995). *Evolutionary computation. Towards a new philosophy of machine intelligence.* IEEE Press.

Forbes, N. (2000). Life as it could be: Alife attempts to simulate evolution. *Intelligent Systems and Their Applications, IEEE*, *15*(6), 2–7.

Francois, P., & Hakim, V. (2004). Design of genetic networks with specified functions by evolution in silico. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(2), 580–585. doi:10.1073/pnas.0304532101

Gilbert, S. F. (Ed.). (2003). *Developmental biology, seventh edition*. Sinaur Associates, Inc.

Jin, Y., & Sendhoff, B. (2008). Evolving *in silico* bistable and oscillatory dynamics for gene regulatory network motifs. In *Proceedings of the Congress on Evolutionary Computation 2008*, Hong Kong (pp. 386-391).

Keller, E. F. (2005). Revisiting "scale-free" networks. *BioEssays*, *27*, 1060–1068. doi:10.1002/bies.20294

Kitano, H. (2002). Computational systems biology. *Nature*, *420*, 206–210. doi:10.1038/nature01254

Knabe, J. F., Nehaniv, C. L., & Schilstra, M. J. (2008). Genetic regulatory network models of biological clocks: Evolutionary history matters. *Artificial Life*, *14*(1), 135–148. doi:10.1162/artl.2008.14.1.135

Kwon, Y. K., & Cho, K. H. (2008). Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics. *Bioinformatics (Oxford, England)*, *24*(7), 987–994. doi:10.1093/bioinformatics/btn060

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A., & Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, *431*, 308–312. doi:10.1038/nature02782

Lynch, M. (2007). *The origins of genome architecture.* Sunderland, MA: Sinaur Associates Inc.

Miller, J. F. (2003). Evolving developmental programs for adaptation, morphogenesis, and self-repair. In . *Proceedings of the European Conference on Artificial Life, ECAL, 2003*, 256–265.

Nicolis, G. (1995). *Introduction to nonlinear science.* Cambridge University Press.

Paladugu, S. R., Chickarmana, V., Deckard, A., Frumkin, J. P., McCormack, M., & Sauro, H. M. (2006). *In silico* evolution of functional modules in biological networks. *IEE Proc.- . Systematic Biology*, *153*(4), 223–235.

Rudge, T., & Geard, N. (2005). Evolving gene regulatory networks for cellular morphogenesis. In *Recent Advances in Artificial Life, Advances in Natural Computation,* 3, 231-252. World Scientific Publishers.

Schwefel, H. P. (1995). *Evolution and optimum seeking*. Wiley & Sons, Inc.

Steiner, T., Olhofer, M., & Sendhoff, B. (2006). Towards shape and structure optimization with evolutionary development. In *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems, ALife X* (pp. 70-76).

Steiner, T., Schramm, L., Jin, Y., & Sendhoff, B. (2007). Emergence of feedback in artificial gene regulatory networks. In . *Proceedings of the Congress on Evolutionary Computation, 2007*, 867–874. doi:10.1109/CEC.2007.4424561

FURTHER READING

Aldana, M., Balleza, E., Kauffman, S., & Resendiz, O. (2007). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology, 245*(3), 433–448. doi:10.1016/j.jtbi.2006.10.027

Alon, U. (2006). *An introduction to systems biology*. Chapman & Hall/CRC.

Babu, M. M. (in press). Introduction to microarray data analysis. In R. Grant (Ed.), *Computational genomics*. UK: Horizon Press.

Bentley, P., & Kumar, S. (1999). Three ways to grow designs: A comparison of the embryogenies for an evolutionary design problem. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela & R. E. Smith (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference*, 1, 35-43.

Bongard, J. (2002). Evolving modular genetic regulatory networks. In *Proceedings of the Congress on Evolutionary Computation*, (pp. 1872-1877).

Bowtell, D. D. (1999). Options available--from start to finish--for obtaining expression data by microarray. *Nature Genetics*, *21*, 25–32. doi:10.1038/4455

Brenner, S., Jacob, F., & Meselson, M. (1961). An unstable intermediate carrying information from genes to ribosomes for protein synthesis. *Nature*, *190*, 576–581. doi:10.1038/190576a0

Cherry, J. L., & Adler, F. R. (2000). How to make a biological switch. *Journal of Theoretical Biology*, *203*, 117–133. doi:10.1006/jtbi.2000.1068

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*(1), 67–103. doi:10.1089/10665270252833208

Di Cera, E., Phillipson, P. E., & Wyman, J. (1989). Limit-cycle oscillations and chaos in reaction networks subject to conservation of mass. *Proc. Natl. Acad. Sci. USA Biophysics*, *86*, 142–146. doi:10.1073/pnas.86.1.142

Drennan, B., & Beer, R. D. (2006). Evolution of repressilators using a biologically-motivated model of gene expression. In *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems (*pp. 22-27).

El-Samad, H., Kurata, H., Doyle, J. C., Gross, C. A., & Khammash, M. (2005). Surviving heat shock: Control strategies for robustness and performance. *Proceedings of the National Academy of Sciences of the United States of America, 102*(8), 2736–2741. doi:10.1073/pnas.0403510102

Elowitz, M. B., & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature, 403,* 335–338. doi:10.1038/35002125

Frank, J. (2000). The ribosome-a macromolecular machine par excellence. *Chemistry & Biology, 7,* 133–141. doi:10.1016/S1074-5521(00)00127-7

Furusawa, C., & Kaneko, K. (2002). Origin of multicellular organisms as an inevitable consequence of dynamical systems. *The Anatomical Record, 268,* 327–342. doi:10.1002/ar.10164

Gurdon, J. B. (1992). The generation of diversity and pattern in animal development. *Cell, 68,* 185–199. doi:10.1016/0092-8674(92)90465-O

Hartwell, L., Hood, L., Goldberg, M. L., Silver, L. M., Veres, R. C., & Reynolds, A. (2000). *Genetics: From genes to genomes.* Boston: McGraw Hill.

Hegde, P., Qi, R., Abernathy, K., Gay, C., Dharap, S., & Gaspard, R. (2000). A concise guide to cDNA microarray analysis. *BioTechniques, 29,* 548–562.

Hogeweg, P. (2000). Shapes in the shadow: Evolutionary dynamics of morphogenesis. *Artificial Life, 6,* 85–101. doi:10.1162/106454600568339

Horak, C. E., & Snyder, M. (2002). ChIP-chip: A genomic approach for identifying transcription factor binding sites. *Methods in Enzymology, 350,* 469–483. doi:10.1016/S0076-6879(02)50979-4

Kaluza, P., Ipsen, M., Vingron, M., & Mikhailov, A. S. (2007). Design and statistical properties of robust functional networks: A model study of biological signal transduction. *Physical Review E,* 75, 015101-1 ff.

Lodish, H., Berk, A., Zipursky, S. L., Matsudaira, P., Baltimore, D., & Darnell, J. (2000). *Molecular cell biology, third edition.* New York: WH Freeman.

Mangan, M., & Alon, U. (2003). Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences of the United States of America, 100*(21), 11980–11985. doi:10.1073/pnas.2133841100

Oikonomou, P., & Cluzel, P. (2006). Effects of topology on network evolution. *Nature Physics, 2,* 532–536. doi:10.1038/nphys359

Schena, M., Heller, R. A., Theriault, T. P., Konrad, K., Lachenmeier, E., & Davis, R. W. (1998). Microarrays: Biotechnology's discovery platform for functional genomics. *Trends in Biotechnology, 16,* 301–306. doi:10.1016/S0167-7799(98)01219-0

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics, 31,* 64–68. doi:10.1038/ng881

Tyson, J., Chen, K. C., & Novak, B. (2003). Sniffers, buzzers, toggles, and blinkers: Dynamics of regulatory and signaling pathways in the cell. *Current Opinion in Cell Biology, 15*, 221–231. doi:10.1016/S0955-0674(03)00017-6

Watson, J. D., Hopkins, N. H., Roberts, J. W., Steitz, J. A., & Weiner, A. M. (1987). *Molecular biology of the gene, fourth edition.* Menlo Park, CA.

Wu, J., Smith, L. T., Plass, C., & Huang, T. H. (2006). ChIP-chip comes of age for genomewide functional analysis. *Cancer Research, 66*, 6899–6902. doi:10.1158/0008-5472.CAN-06-0276

Yi, T.-M., Huang, Y., Simon, M. I., & Doyle, J. C. (2000). Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proceedings of the National Academy of Sciences of the United States of America, 97*(9), 4649–4653. doi:10.1073/pnas.97.9.4649

KEY TERMS AND DEFINITIONS

Gene Regulatory Network: A gene regulatory network (also called a GRN or genetic regulatory network) is a collection of DNA segments in a cell which interact with each other and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed.

Evolution: The historical development of a biological group (as a race or species).

Phylogeny: A theory that the various types of animals and plants have their origin in other preexisting types and that the distinguishable differences are due to modifications in successive generations ; also: the process described by this theory.

Development (Biology): The formation and growth of the embryo to a mature state.

Feedback: The return to the input of a part of the output of a system, or process (as in an automatic control device that provides self-corrective action).

Evolution Strategy (ES): A specific variant of evolutionary algorithms originally focusing on the phenotypic level of evolution and adaptation, ES was developed by Rechenberg, Schwefel and coworkers [1].

Robustness (Biology): Capability of performing without failure under a wide range of conditions / under genetic mutation.

Genetic Switch: Part of a GRN that creates a shift from one expression state of a gene to another, depending on a chemical signal.

Chapter 22
A Model for a Heterogeneous Genetic Network

Ângela T. F. Gonçalves
Darwin College, UK

Ernesto J. F. Costa
Pólo II- Pinhal de Marrocos, Portugal

ABSTRACT

In this chapter, we propose a new model for gene regulatory networks (GRN). The model incorporates more biological detail than other approaches, and is based on an artificial genome from which several products like genes, mRNA, miRNA, noncoding RNA, and proteins are extracted and connected, giving rise to a heterogeneous directed graph. We study the dynamics of the networks thus obtained, along with their topology (using degree distributions). Some considerations are made about the biological meaning of the outcome of the simulations.

INTRODUCTION

The recent ability to sequence an organism's genome, in particular the human one, was a great breakthrough thought to be the key to new ways to diagnose, treat, and some day prevent the thousands of disorders that affect us. However, simply knowing the gene sequences is not enough and the challenge is currently in deciphering how genes determine the phenotypic traits of an organism and how the genome controls the development of organisms.

For a long time it was believed that the DNA in the genes was transcribed into RNA, which in turn was translated into proteins in a one-way process. This is called the molecular biology's central dogma. The central dogma explains the basic process of gene expression into proteins, but is unable to explain several essential phenomena such as cellular differentiation, where cells with the same genetic information to behave differently according to their function in the organism. The explanation to such unaccountable

DOI: 10.4018/978-1-60566-685-3.ch022

processes lies in complex networks of interactions, known as regulatory networks, between genes and many other molecules including proteins, the very products of gene expression.

Since these regulatory networks are highly non-linear and have several thousand variables it is paramount to find computational models for them, albeit the difficulty of the task (Voit, 2000). Various approaches more or less abstract and more or less general for modeling gene regulatory networks appeared in the last decades. On one end of a possible axis of classification there are highly detailed biochemical models, such as Reinitz's phage-λ model (Reinitz, 1990), with which predictions and simulations of small and well understood systems can be performed; on the other end there are abstract models, such as Stuart Kauffman's Random Boolean Networks (Kauffman, 1993), that are commonly used to study broad scale and gross properties of the networks. Towards this more abstract end of the range, considerations are usually made about the network's topologies and dynamics (Kauffman, 1993; Thomas, 2001; Liebovitch, 2006; Barabási, 1999; Reil, 1999; Kuo, 2004; and Banzhaf, 2003; to name just a few). Most of the models used in these studies often merge the several processes that occur in protein synthesis, or focus on regulation only at transcription level. Given that regulation occurs at any stage of protein synthesis including transcription, RNA processing, mRNA decay, translation and post-translation (Hartl, 2005); it should be interesting to observe how the dynamics and topologies might be different when intermediate steps and entities are considered.

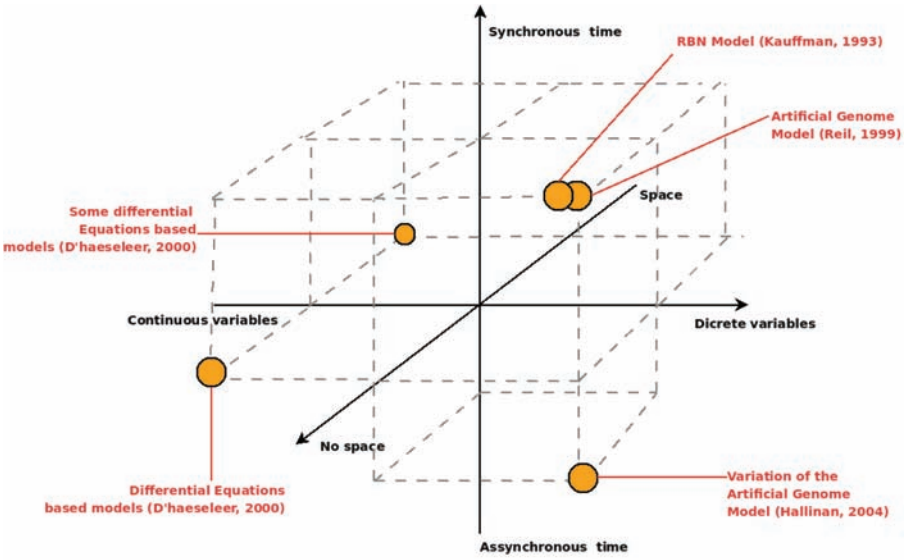
In this chapter we consider existing classes of models and their relevance for the exploration of theories and hypothesis regarding the structural and dynamic properties of regulatory networks when additional layers of regulation are taken into account. A new model for gene regulation complying with this aim is described later on. In order to compare the networks obtained with this model with networks obtained from previous models, we study some of their statistical properties, including: topology (using degree distributions) and dynamic behaviors.

BACKGROUND

Several models for Gene Regulatory Networks have been proposed in recent years. Because the biological processes involved in gene regulation are so highly complicated, the majority of these makes the assumption that the control of gene expression resides only in the regulation of gene transcription. Moreover, this may also be due to the nature of the most widely available microarray data (Geard, 2004; D'haeseleer, 2000). This overview is not meant to be exhaustive and we only briefly mention some of the known models. For a more extensive review and in-depth descriptions see de Jong (2002), Hasty (2001), Goncalves (2007), D'haeseleer (2000) and Geard (2004). We can classify the models that will be discussed here according to the following aspects: variables such as product concentrations are discrete, continuous or mixed; time is discrete and the update of the variables is either synchronous or asynchronous (there are, however, cases where time is continuous); space is discrete, continuous or absent (see Figure 1).

One very influential discrete approach early adopted a complex systems view of the genome (Kauffman, 1993). In this approach Kauffman represented the regulatory system as a network of logical components connected at random, creating networks, which he coined as Random Boolean Networks (RBNs). These RBNs exhibited emergent properties, such as cyclic attractors, point attractors, robustness and homeostasis, that also occur in real biological systems. The abstract similarity between the RBNs and biological cells made the simplification of modelling time as discrete time steps and considering only

Figure 1. A classification of regulatory network models. Some of these models are used with evolutionary algorithms, such as Genetic Algorithms, which have evolutionary time (iterations of the algorithm). Those models were placed at the origin of the time axis because the synchronous or asynchronous property is related to the timing of the variable update.

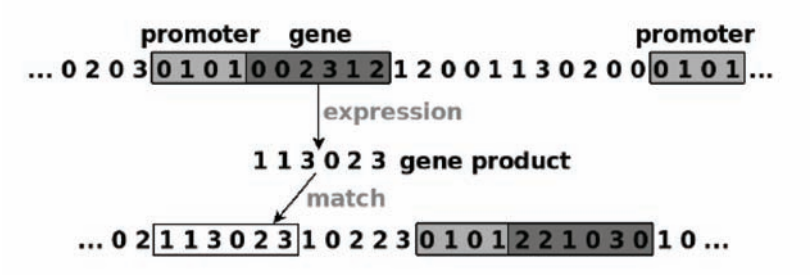


binary levels of gene activity (on or off) widely accepted. However, despite the interesting insights of Kauffman's model it was unable to give much explanation for the regulatory mechanisms and, to many, did not exhibit sufficient parallels with the real networks (Reil, 1999). Rather than using a network for a base level representation, another discrete model was proposed by Reil (1999) using a more biological framework. This model, called the Artificial Genome, was based on a DNA-like sequence representing the genome, from which the network structure could be extracted. Similar models were proposed by Banzhaf (2003) and described by Hallinan (2004), Watson (2004) and Willadsen (2003).

In the Artificial Genome model a random sequence of bases is generated to represent an organism's genome. This sequence is searched for genes, each one being defined by the six digits that follow each '0101' (defined to represent the promoter) sequence found. The sequences between genes are the regulatory regions for the genes immediately following them and the expression process is represented by incrementing each gene's digits by one, modulo 4 (the number of bases). The sequence resulting from this operation is said to be the gene product and it will be used to search for matches in the regulatory regions. Whenever there is match, a regulatory link between the gene that originated the gene product and the gene regulated by the region where the match occurred is created (Figure 2). The regulation will be inhibitory or excitatory depending on the value of the last digit of the gene product and, after searching for all possible matches, a regulatory network can be created and displayed in a graph.

The Artificial Genome and its variations aimed to be biologically more significant by integrating developmental biology, providing in this way a richer ground for comparison. Similarly to the RBNs, the Artificial Genome presented different behaviors depending on the connectivity. With low connectivity the system enters a frozen regime, with high connectivity a chaotic regime and with medium levels the system enters the critical regime displaying cyclic patterns of expression. However, one of the criticisms

Figure 2. The artificial genome model. © 2008 MIT Press. Used with permission.



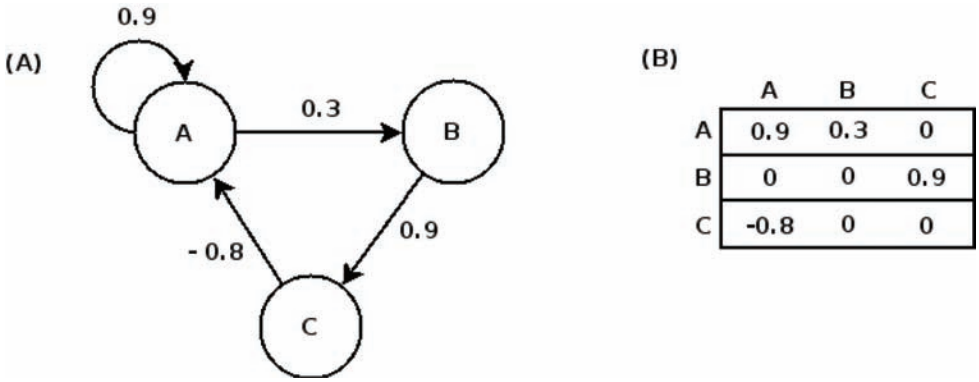
about RBNs was that the nodes were only allowed a low connectivity in order to produce the different dynamics. In fact, with a connectivity greater than 2, the RBN networks would generally show chaotic dynamics whereas, in the Artificial Genome this value was found to be significantly higher (Willadsen, 2003). Other works on this model focused on the study of the effects of sequence-level mutations at the network level (Watson, 2004) and others still, observed that there is a significant amount of unused sequence in genomes with complex gene expression (Reil, 1999). However, despite the many interesting insights, the Artificial Genome still comprises many simplifications. The main one being the merging of the entire process of gene expression (ignoring the intermediate products) and the use of an arbitrary operation for the creation of a gene product.

While the above models represent variables discretely, all product concentrations, activation levels or rates of transcription of genes can, in reality, vary in a continuous fashion. As such, several models have been proposed that take variables as continuous values and use differential equations to determine them. Examples of such models are the Additive Regulation Model, Neural Network models and models based on the S-System power law.

In the Additive Regulation Model (D'haeseler, 2000) the regulatory relations between genes are represented in a matrix of positive, negative or zero connections (Figure 3).

The occurrence of a nonzero entry in the matrix at row *i*, column *j*, indicates that there is a regulatory

Figure 3. (A) Graph of a regulatory network where gene A regulates itself and gene B. Gene B regulates gene C which in turn regulates gene A. (B) Matrix representation of the graph. © 2008 MIT Press. Used with permission.



connection from gene product *i* to gene product *j*. If this entry is positive the regulation is enhancing and repressive if it is negative. The expression level of each gene (x_i), updated synchronously in each time step, could be given by the weighted sum of all variables:

$$\frac{dx_i}{dt} = \sum_j w_{ji} x_j + b_i$$

with x_j the expression level of the *i*th gene, b_i a bias term that indicates if the gene is expressed in the absence of regulatory inputs and w_{ji} the weight in the matrix from gene *j* to gene *i* . The previous equation could be improved to include the observation that most genes may have a sigmoidal response curve, i.e. the gene activation increases slowly and saturates at a maximum level. If, furthering the biological plausibility, a decay rate of gene products was added, the expression would become: $\frac{dx_i}{dt} = S\left(\sum_j w_{ji} x_j + b_i\right) - D_i x_i$ with *S()* a sigmoidal function and D_i the decay rate of gene *i*. The introduction of the nonlinear function, which should in principle be able to match the underlying regulatory network more closely, makes this model now similar to a recurrent neural network.

Regarding neural networks, some researchers investigated variations of the basic model, including a model in which proteins and mRNA products are represented by separate network layers (Vohradský, 2001). It is possible to create a mapping between a neural network and a system of ordinary differential equations. The advantage of using such a model instead of the general-purpose differential equations is the availability of well known efficient learning algorithms (D'haeseler, 2000). Learning involves adjustments to the weights of the links that exist between the nodes of the network.

Finally, a model based in S-systems could be defined by a parameterized set of nonlinear differential equations:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{n} x_j(t)^{g_{ij}} - \beta_i \prod_{j=1}^{n} x_j(t)^{h_{ij}}$$

with x_i the expression level of gene *i*, *n* the number of network components, $\alpha_i \geq 0$, $\beta_i \geq 0$ rate constants and g_{ij}, h_{ij} a representation of the interactive affectivity of x_j to x_i. The first product in the equation describes all the excitatory influences (influences that increase x_i), while the second product describes all the inhibitory ones (that decrease x_j).

These systems have a rich structure and should be flexible enough to capture relevant dynamics (Savageau, 1976; Voit, 2000; Kikuchi, 2003; and others), but the number of parameters that have to be estimated is large (about $2n(n+1)$) (Noman, 2005).

With the continuous models described above, biologically plausible features can be included (D'haeseleer, 2000) and reverse engineering/learning algorithms used to determine their parameters from real data (Ando, 2001; Sakamoto, 2001; Noman, 2005; van Someren, 2002). However, as with the previous discrete models discussed, the black box approach of the process they use makes them less suitable for the understanding of the mechanisms of gene regulation at the various levels.

These considerations motivated the creation of a new model with a string based framework similar to the Artificial Genome but breaking the expression process down to some important steps and overcoming some of its simplifications (Goncalves, 2007). The networks derived by this model, called HeRoN, can be represented by a graph where the nodes represent the different products involved in gene expres-

sion, thus heterogeneous, and the arcs establish the interactions between them. These networks can be analyzed from the topological and dynamical point of view of complex systems.

A MODEL FOR A HETEROGENEOUS GENETIC NETWORK

The model we propose, HeRoN, is based on a string, from a four symbol alphabet, that represents a genome and derives from it various products such as genes, proteins and some more intermediate products. The expression algorithm is a six-step process:

1. Generate the genome.
2. Search the genome for genes.
3. Generate RNA transcripts from the genes.
4. Splice the RNA transcripts.
5. Translate the mRNA into proteins.
6. Generate microRNAs from the non-coding RNAs.

These steps will now be described in more detail.

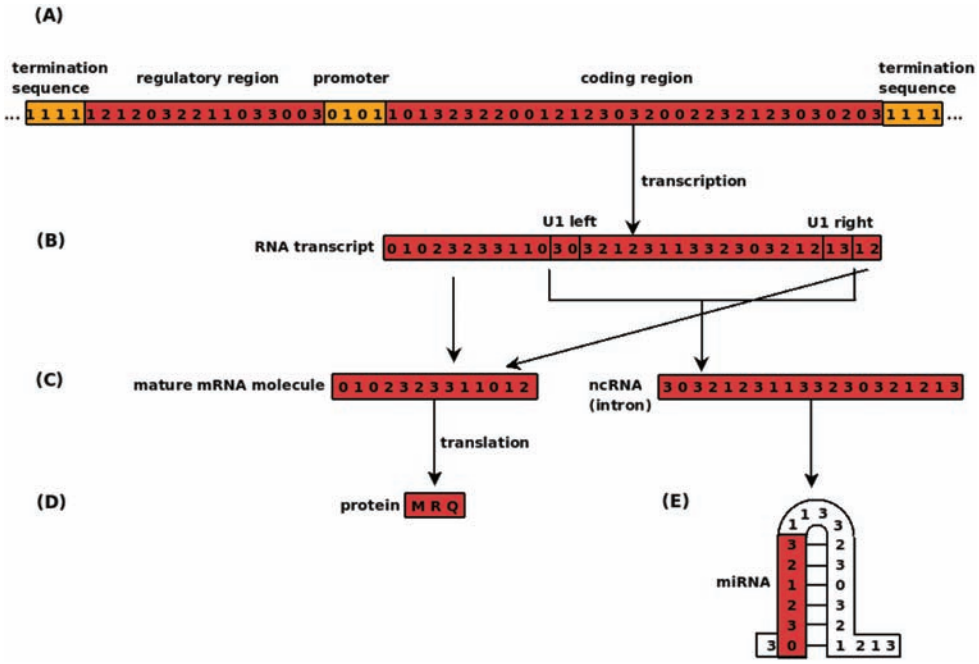
1. Generate the Genome

The genome, represented as a string of integers, is randomly generated with a parametrical size. Each integer corresponds to a base: 0 to Thymine or Uracil, 1 to Adenine, 2 to Guanine, 3 to Cytosine.

2. Search the Genome for Genes

In real biological systems there are some promoter sequences that appear in most genes of many organisms, called consensus sequences, and the more a sequence in a genome resembles them, the more efficient the transcription. In our model the genome is searched for given sequences that represent these gene promoters (e.g. we use the string ‘0101’ to represent the TATA box sequence). A threshold symbolizing the binding strength between a RNA polymerase and the genome, was set as a parameter is such a way that that a sequence in the genome, with the same size as the given promoter sequence (‘0101’), is considered to be a valid promoter when its percentage of match with the given sequence is equal or above this threshold. Each time a valid promoter is found the genome is searched for a termination sequence. When such termination sequence, chosen to be a poly-A sequence of adjustable size (‘AAAA…’ corresponding to a sequence of 1’s in the four letter alphabet), is found a gene is created. Each gene consists of a promoter sequence, a coding sequence and a regulatory region. The coding sequence is the region located between the promoter and the termination sequence, while the regulatory sequence is the region located between the end of the previous gene (after its termination sequence) and the promoter (Figure 4A).

Figure 4. (A) Structure of a gene in the HeRoN model. The sequences '0101' and '1111' correspond to the TATA box and the poly-A termination sequence, respectively. (B) RNA transcript created by complementing the coding region of the gene. The U1left and U1right sequences highlighted are the two consensus sequences that signal the presence of an intron. (C) On the left a mature mRNA molecule is created by joining the two ends of the original sequence after the intron (represented on the right) is removed. (D) Protein created after the start codon '102' which represents AUG was found on the mRNA molecule (translation begins at its second base where the start codon is located). Translation ends when a stop codon is found. In this example the stop codon '012', which represents UAG, is located at the mRNAs eleventh base. (E) miRNA molecule generated from an intron with a stem-and-loop motif. © 2008 MIT Press. Used with permission.



3. Generate RNA Transcripts from the Genes

RNA transcripts are generated by complementing the bases in the coding sequence of the genes according to the pairing A-T and C-G. In the four-integer alphabet the complementary pairs are 1-0 and 3-2 (Figure 4A-B).

4. Splice the RNA Transcripts

Each RNA transcript is searched for introns that are removed from the sequence and stored in a list of components called non-coding RNAs (ncRNAs). The introns are detected by means of two sequences, called U1left and U1right, that simulate the role of U1 srRNA molecule which has two highly conserved consensus sequences complementary to the 5' and 3' ends of essentially all mRNA introns (Zhang, 1999). The new sequences created from the RNA transcripts with the introns removed are called mRNA (Figure 4B-C).

5. Translate the mRNA into Proteins

The mRNA molecules are searched for a start codon sequence (AUG, corresponding to '102' in the four integer alphabet). When this sequence is found the mRNAs are read three bases at a time until a stop codon is found (UAA, UGA, or UAG, corresponding to '011', '021' or '012'). Each three bases are translated into one amino acid according to the genetic code table. The stop codon is not considered part of the protein (Figure 4C-D).

6. Generate microRNAs from the Non-Coding RNAs

This step was added to the model because a large number of RNA transcripts did not produce proteins due to the fact that they missed the start codon. Searching in the literature for similar phenomena led to the subject of non-coding/junk DNA. Junk DNA has been a name given by researchers to large regions of DNA for which no function has yet been found. These regions include introns and large portions of intergenic sequences. Having found evidence that genes considered to be junk DNA have a regulatory influence (Martens, 2004) and that this kind of DNA makes up to 95% of chromosomes, researchers changed its name to non-coding DNA. In particular, the regulatory role of noncoding genes relates to the RNA interference (RNAi) mechanism. This mechanism of transcriptional gene silencing is induced by molecules of RNA associated with proteins. These molecules are called small interfering RNA (siRNA) when they derive from exogenous sources (outside the cell), or microRNA (miRNA), when they are produced from non-coding genes in the cell's own genome. miRNAs are short single-stranded RNA stretches of 21 to 23 nucleotides processed from primary transcripts known as pre-miRNA to short stem-loop structures called pre-miRNA and finally to functional miRNA (Gregory, 2006). The effect of their regulatory mechanism is that while some genes are transcribed at a normal rate, they are not expressed into proteins because they are degraded before leaving the nucleus.

The influence of these microRNAs was included in the model by defining that if a mRNA does not produce a protein, because it misses the start codon, that mRNA molecule is considered to be non-coding and therefore is added to the ncRNA list where the introns were already stored. All ncRNAs are then scanned for hairpin loops with a minimum length. This indicates the presence of miRNAs that are considered as another product in the model (Figure 4C-E).

This expression algorithm just described creates lists of products and stores their corresponding sequences. References to the products from which they originated are also kept. To determine the interaction network between the different products it is necessary to determine how they bind to one another, namely proteins to genes and miRNAs to genes.

Finding the interactions between miRNAs and genes is straightforward since the two products are made of the same units, nucleotides, and their binding is a simple match between complementary sequences. The other type of bindings involves elements that do not interact in a linear manner and are made up of different units, amino acids and nucleotides. In biological systems the protein's ability to locate and bind with certain DNA sequences depends not only on the involved amino acid and nucleotide sequences but also on the protein's three-dimensional structure and on the DNA's double stranded structure.

Many models exist that try to predict DNA-protein binding sites (Baker, 2001) yet this is still an open topic in Bioinformatics. In addition to the existing approaches some authors find it important to examine the individual interactions between the amino acids and the nucleotides since underlying the binding are

the discrete interactions between them (Hoffman, 2004; Luscombe, 2001). Databases such as the Amino Acid-Nucleotide Interaction Database (AANT) categorize amino-acid-nucleotide interactions from experimentally determined protein-nucleic acid structures. In our model we use the statistical table of the entire AANT database (Table 1) along with a binding threshold to determine if a given protein binds with a DNA sequence. For each amino acid in the protein its binding probability with the corresponding nucleotide in the DNA is given by the AANT statistic table. The interactions are compared with the threshold using one of four methods called: average, maximum, minimum and random (Figure 5).

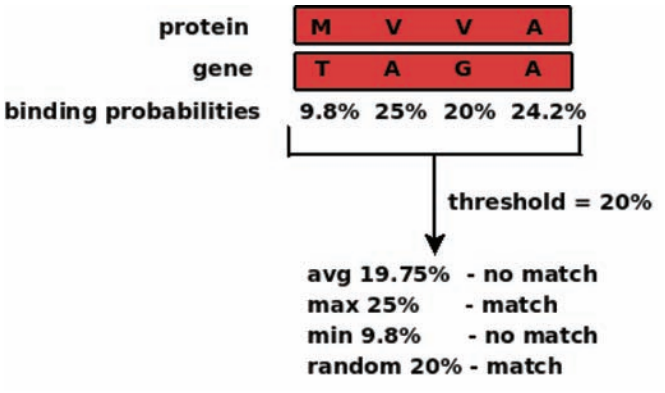
When using the average ('avg') method an average of all the probabilities is calculated. For the maximum ('max') and minimum ('min') methods, the respective maximum or minimum probability is chosen. For the random method the probability of a random amino-acid-nucleotide pair, from the sequence, is chosen. In this example it was the V-G pair. The components are said to bind if the resulting probability is above or equal to the threshold.

The information gathered about the interaction between the components can then be used to create a graph representation of the network. In this graph: each gene creates many products, most of them ncRNAs and a single mRNA molecule; each mRNA either creates a protein or a miRNA molecule, and miRNA molecules can also derive from ncRNAs. All connections starting at a miRNA molecule end at a mRNA molecule and are repressive, while connections between proteins and genes can be either activating or repressive (Figure 6).

Table 1. Statistical table of the entire AANT database. Along with the name of the amino acids are the conventional three-letter and one-letter abbreviations. © 2008 MIT Press. Used with permission.

Amino-acid	**A(%)**	**C(%)**	**G(%)**	**T(%)**
Alanine (Ala, A)	24.2	17.3	24.0	24.6
Arginine (Arg, R)	19.6	24.1	35.7	12.2
Asparagine (Asn, N)	25.5	20.0	23.9	17.7
Aspartate (Asp, D)	13.3	34.2	37.0	1.5
Cysteine (Cys, C)	29.1	18.8	24.8	23.1
Glutamine (Gln, Q)	28.0	17.7	29.4	13.7
Glutamate (Glu, E)	19.1	34.8	33.0	4.8
Glycine (Gly, G)	20.1	22.9	32.1	17.0
Histidine (His, H)	25.3	16.2	37.7	14.2
Isoleucine (Ile, I)	21.4	26.4	30.8	11.4
Leucine (Leu, L)	9.5	31.1	30.2	19.4
Lysine (Lys, K)	23.7	22.8	30.7	16.3
Methionine (Met, M)	22.1	27.9	22.1	9.8
Phenylalanine (Phe, F)	17.7	24.1	40.5	17.7
Proline (Pro, P)	37.0	11.0	21.0	2.0
Serine (Ser, S)	28.2	20.9	27.2	19.7
Threonine (Thr, T)	24.6	20.2	27.8	23.1
Tryptophan (Trp, W)	14.4	30.2	24.8	21.8
Tyrosine (Tyr, Y)	28.4	27.4	23.6	15.0
Valine (Val, V)	25.0	35.3	20.0	1

*Figure 5. Each amino-acid-nucleotide pair is searched for in the AANT statistic table (***Table 1***). © 2008 MIT Press. Used with permission.*



EXPERIMENTAL STUDY

The experimental study that was carried out involves two main aspects: the study of the structural properties of the graphs and their dynamics.

The variable parameterization used throughout the experiments performed is shown in Table 2 and the parameters that were kept fixed are: string '0101' for the promoter, promoter match threshold of at least 75%, string '1111' (four base long poly-A) for the termination sequence, and a binding site size of 6 for the proteins. Experiments were run for all possible combinations of the 'used values' column on Table 2 and each combination of the variable parameters was run 10 times. The initial set of active genes for each of the runs was randomly taken from a uniform distribution.

Figure 6. Activation/deactivation relationships between the different products. The positive and negative signs near the edges represent, respectively, activation or deactivation of a product. The black colored edges represent the regulatory connections while the grey edges represent the creation of a product. © 2008 MIT Press. Used with permission.

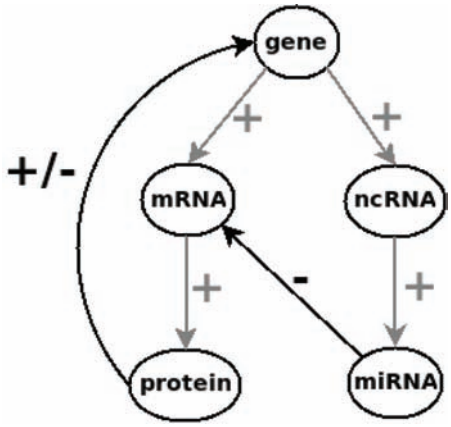


Table 2. Variable parameterization. © 2008 MIT Press. Used with permission.

Parameter	Used values	Possible values
genome size	20000, 100000 and 500000	any positive integer
miRNA binding site size	4, 5, 6 and 7	any positive integer
inhibition rate	0, 0.25, 0.50 and 0.75	any value between 0 and 1
binding threshold	29, 32, 33 and 34	any positive integer
binding choice	avg, max, min and rand	avg, max, min or rand

Topology of the Networks

The topology of a network is its most basic feature and three different classes of networks, regular lattice, small-world and random networks, arise from the different ways large sets of elements connect. A network where each node is connected to its nearest spatial neighbors is called a regular lattice. Starting with a regular lattice and randomly rewiring a portion of its links creates small-world networks. At the extreme of this rewiring random networks are formed.

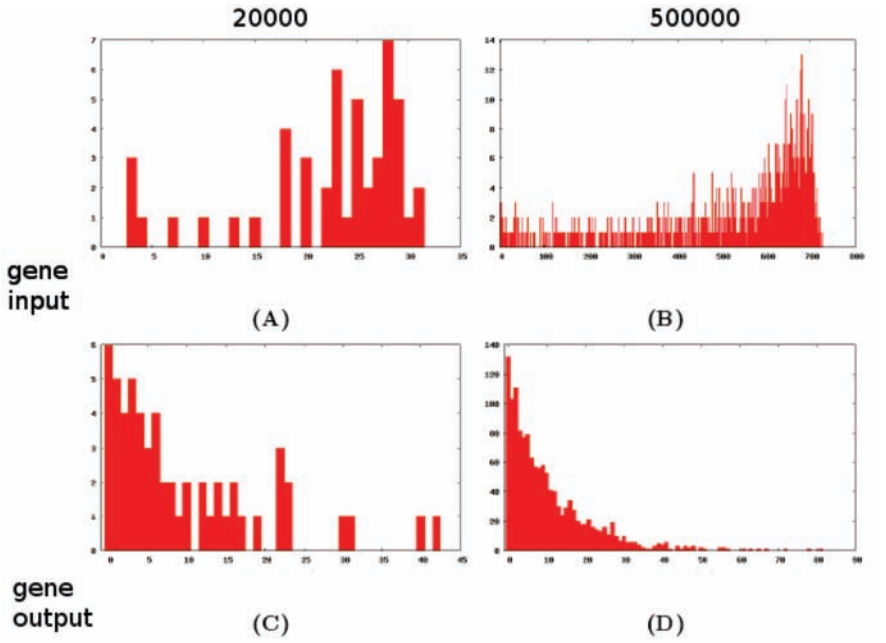
Gene regulatory networks, like most social and biological networks, such as the World Wide Web, the immune system, the brain and ant colonies, to name just a few, possess certain topological features that are non-trivial. For instance, while nodes on regular lattices have constant degree and ordinary random networks have Poisson degree distributions, it is found that many real-world networks have degree distributions measurably different from these. This strongly suggests that there are features of such networks that would be missed if they were to be approximated by an ordinary random graph or lattice (Newman, 2001), thus many recent works on real-world complex systems focus on the subject of small-world and scale-free networks. However, while there are several statistical properties of graphs that can be used to characterize their topology (e.g., the average path length or the clustering coefficient), the work done on this model focuses on the node's degree distributions.

With very few exceptions (Newman, 2001), most frameworks for the study of graph statistical properties have been developed for unipartite, undirected graphs, i.e. for graphs with a single type of nodes, as opposed to *n*-partite graphs that have *n* distinct sets of nodes with undirected edges between them. It is, however, an important aspect for us to consider directed and heterogeneous graphs, since this is the case of the network graphs obtained with our model. One consequence of the graph being directed is that nodes have two different kinds of edges, the ones arriving to it and the ones leaving it - these will be referred to, respectively, as input and output connections. This is particularly important in analyzing the degree distribution of the nodes and therefore nodes of different kinds will be analyzed separately in relation to input and output connectivity.

The input and output degree distributions for each kind of node for a 20,000 base long genome and for a 500,000 base long genome are shown in Figure 7 and Figure 8, respectively on the left and right columns. Each column refers to the same network, obtained with a binding threshold of 29, the 'avg' binding choice, a miRNA binding size of 6 and an inhibition rate of 0. The number of products of each kind in networks of different sizes (with the same parameterization as above) is given in Table 3.

When comparing the columns in Figure 7 and Figure 8 it can be noticed that the 'genome size' parameter does not seem to qualitatively alter the connectivity distributions. However, the two parameters 'binding threshold' and 'binding choice' determine the input connectivity distribution of the genes and

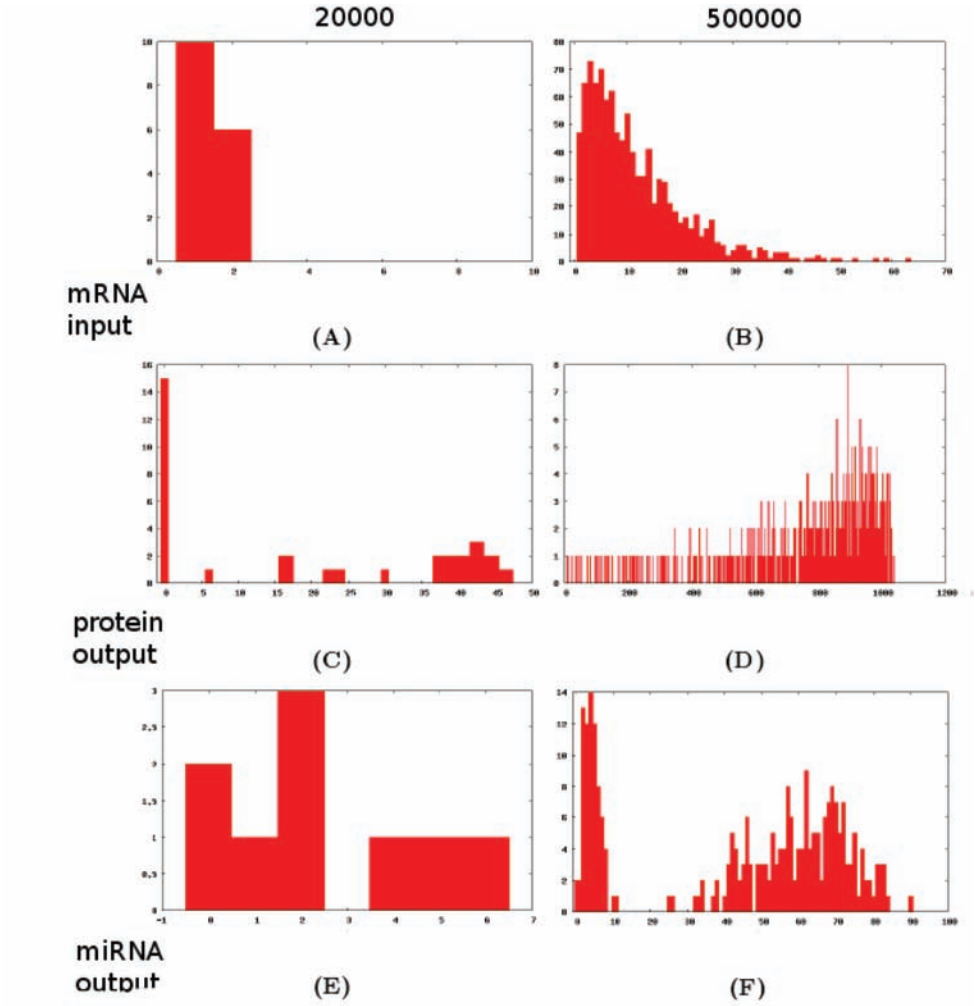
Figure 7. Histograms of the degree distribution of gene input and output connectivities for a 20,000 base long genome (left column), and a 500,000 base long genome (right column). (A) and (B) Gene input connectivity distribution. (C) and (D) Gene output connectivity distribution. © 2008 MIT Press. Used with permission.



the output connectivity distribution of the proteins. With a low binding threshold of 29 and a 'max' or 'avg' binding choice, a heavily left skewed distribution with a fat tail is found for both the gene input and protein output connectivity (Figure 9A and C). These distributions are consistent with studies on other complex systems represented by directed graphs (Newman, 2001). For higher binding thresholds of 32, 33 and 34, input and output connectivity distributions change shape (Figure 9B and D) and display a right side tail. Only the 'max' and 'rand' binding choices keep showing the input and output connectivity distributions of Figure 9A and C. Both the 'max' and 'rand' binding choices produce similar gene input and protein output connectivity distributions but the tail of the 'max' is shorter. This difference may not be significant to the observable dynamic behavior of the network, and in fact it is not, as shall be discussed later.

Regarding the gene output connectivity the shape of the distribution (Figure 7D) is always maintained because the parameters that could alter it (promoter, promoter match, termination sequence, left and right u1 and u1 match) were kept unchanged throughout the experiments. The linear-log and the log-log plots for the gene output can be seen in Figure 10. On the linear-log plot the distribution falls on a straight line, indicating an exponential decay of the distribution of connectivity. On the log-log plot the distribution decays faster than a power law would, since if the distribution had a power law tail it would fall on a straight line. This is consistent with the work of some authors, who have shown evidence for the occurrence of three classes of small-world networks in real world networks: scale-free networks, characterized by a vertex connectivity distribution that decays as a power law; broad-scale networks, characterized by a connectivity distribution that has a power law regime followed by a sharp cutoff like

Figure 8. Histograms of the degree distributions of the different species for a 20,000 base long genome (left column), and a 500,000 base long genome (right column). (A) and (B) mRNA input connectivity distribution. (C) and (D) Protein output connectivity distribution. (E) and (F) miRNA output connectivity distribution. © 2008 MIT Press. Used with permission.



an exponential or Gaussian decay of the tail; and single-scale networks, characterized by a connectivity distribution with a fast decaying tail, such as exponential or Gaussian. The question of why this range of possible structures for small-world networks exists is explained by the preferential attachment of new nodes that gives rise to power law distributions. Consequently, in the broad-scale and single-scale networks there must be constraints limiting the addition of new links (Amaral, 2000).

In our model, one constraint exists for the connection of new nodes to genes that may account for the faster decay of the tail of the gene output distribution. Genes have output connections to two different types of nodes: mRNA nodes and ncRNA nodes. While a gene only produces one mRNA, it can produce several ncRNAs and, as such, those connections are the most significant in terms of the overall degree distribution. Bigger genes have higher probability of producing several ncRNAs but their ”ability” to

Table 3. Number of each kind of product for different genome sizes, binding threshold 29 and binding choice 'avg'. © 2008 MIT Press. Used with permission.

Genome size	Number of genes	Number of mRNAs	Number of ncRNAs	Number of proteins
20000	57	46	556	46
100000	253	194	2993	194
500000	1361	1043	14531	1043

do so decays each time a ncRNA is created, since the sequence being searched is shortened (search for ncRNAs continues after the last found ncRNA).

As with the output of the genes, mRNAs receive two kinds of inputs: each mRNA receives one single input from a gene and possibly several inputs from miRNAs, so the shape of the distribution (Figure 8B) depends mainly on the miRNAs. Parameters that influence the input distribution connectivity of the mRNAs are the genome size and the miRNA binding site size. Figure 11 shows the linear-log and the log-log plots of the mRNA input frequency. Similar to the output connectivity of genes, the linear-log plot falls on a straight line, with an exponential decay and the log-log decays faster than a power law would, therefore indicating that there may be constraints limiting the addition of new links between the miRNAs and the mRNAs. Since bigger mRNAs have higher probability of having more inputs, the shape of the input distribution may be greatly influenced by their size. These considerations indicate that

Figure 9. Histograms giving the gene and protein degree distributions for a binding threshold of 29 and 'avg' binding choice on the left column, and for a binding threshold of 32 and 'avg' binding choice on the right column. (A) and (B) Gene input connectivity. (C) and (D) Protein output connectivity.

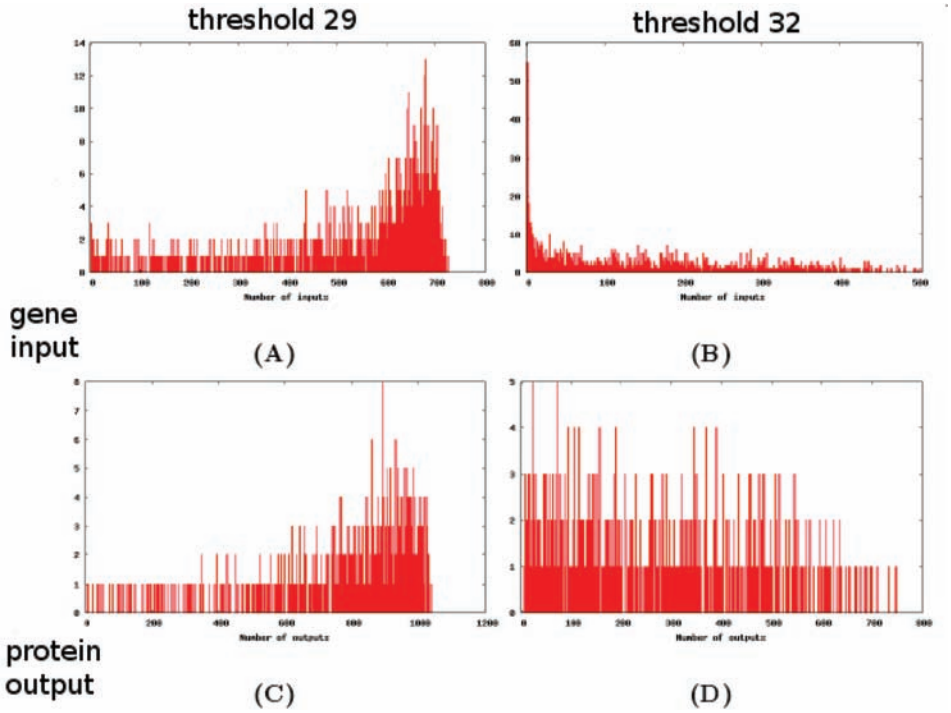
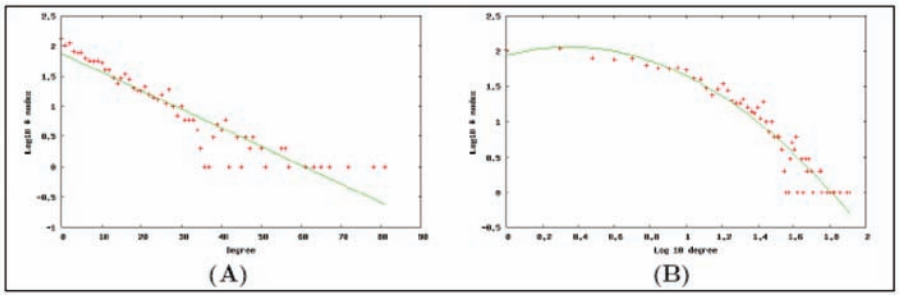


Figure 10. (A) Linear-log plot of the gene output connectivity. (B) Log-log plot of the gene output connectivity. © 2008 MIT Press. Used with permission.



the nature of these networks might not be scale-free as has been hypothesized (Geard, 2004; Hallinan, 2004; Watson, 2004; Willadsen, 2003).

Dynamics of the Networks

Once a regulatory network is obtained its dynamics can be studied by defining a state variable for each element and observing how it changes over time. Starting with all elements 'off', meaning that, regarding genes, they are not being expressed, or that, regarding other products, they are not present, a small number of genes are set to 'on'. This propagates through the network according to the algorithm below in which each element's status is updated synchronously in discrete time steps:

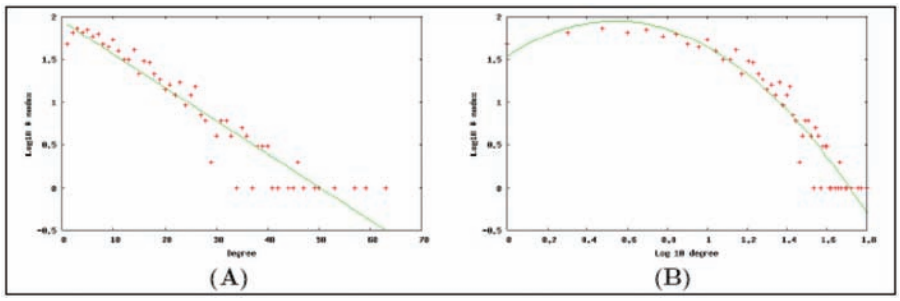
1. Some genes are activated.
2. For each active gene the mRNA and ncRNA molecules that derive from them are activated.
3. For each active mRNA or ncRNA molecule the miRNA and proteins that derive from them are activated.
4. For each active miRNA the genes that are regulated by it are deactivated.
5. For each active protein the genes that are positively/negatively regulated by it are activated/deactivated. If a gene is both activated and deactivated negative regulation takes precedence.
6. Return to 2.

Any active element turns inactive in one time step if its activator element is not active. The status of each kind of product is updated every three steps due to the propagation to the other products.

By being dynamic systems, these networks can be represented by states and transitions and different classes of behaviors can occur, such as the system reaching a point attractor or ordered phase, the system oscillating between two or more point attractors or changing erratically between states with no regular pattern. By an oscillatory behavior it is meant that a cycle of expression repeats itself periodically and this is of particular interest as it was observed that this is an emergent dynamic exhibited by biological gene regulatory networks such as the one described by Hirata (2002).

The three different regimes, ordered, cyclic and chaotic, were found by changing the parameters of the model and some typical examples are depicted in Figure 12. An ordered expression pattern in which some products become permanently active after a short transition period can be seen on the top left of

Figure 11. (A) Linear-log plot of the mRNA input connectivity. (B) Log-log plot of the mRNA input connectivity. © 2008 MIT Press. Used with permission.

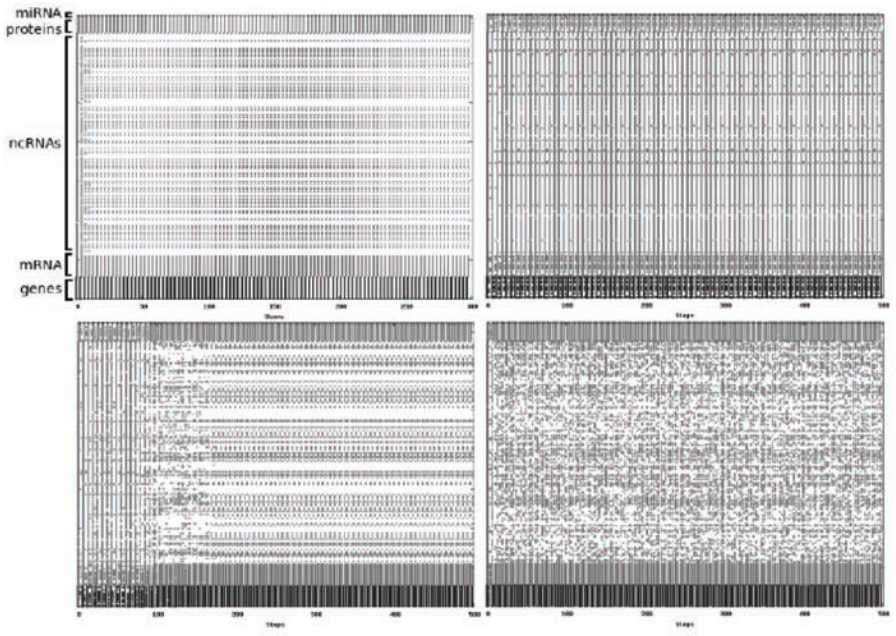


the figure. The cyclic patterns on the top right and bottom left were produced by the same network with different starting conditions (different initial set of active genes), i.e. they are different attractors of the system. On the bottom right a chaotic pattern is shown. The emergence of the three patterns is highly influenced by the variable parameters described in Table 2. Several observations can be made from the results:

- With a low miRNA binding site size there are no significant dynamics. This happens because many miRNAs are created and these only have a repressive effect on the network. Repression is at a rate between 55% and 100%;
- The 20,000 base long genome displays ordered and cyclic dynamics but not chaotic. This could be caused by the network being too small, hence too simple to show that kind of dynamic, or else because the cycle length is shorter. If it is the case of the network being too simple, a network with a 4,600,000 base long genome, which is the size of the E.Coli bacteria, could be expected to behave differently;
- The 'max' and 'random' binding choice parameters display the same dynamics. This is an interesting observation that can be related to the topologies they create, which are only slightly different in respect to the connectivity distributions of protein outputs and gene inputs;
- The 'max' and 'random' binding choice parameters only display ordered dynamics. It can be observed from their connectivity distributions that these choices create networks with more edges between proteins and genes than the 'avg' and 'min' choices. They are less influenced by the binding threshold parameter. Because they influence the creation of regulation connections between proteins and genes they are also less affected by the inhibition derived from the miRNAs.
- Cyclic and chaotic behaviour only appear with some amount of protein caused inhibition. There is a slow transition from ordered dynamics, through cyclic dynamics, to chaotic dynamics with the increase of the inhibition rate of the proteins. Another observation is that, for the 500,000 genome, with high thresholds (above 29), high protein inhibition (above 50%) and the average binding choice, all networks display chaotic behaviour for high values of miRNA size (6, 7 and 8).

In general it was observed that higher values of the miRNA size decrease the amount of inhibition in the network. For instance, while with miRNA size 5 and 50% protein inhibition rate, the total inhibition on the network (miRNA + protein inhibition) is between 73% and 93% (increasing with higher values of the threshold), with higher miRNA sizes the total inhibition is between 50% and 83% (the miRNA

Figure 12.



influence is weaker). The increase in total inhibition rate with higher values of the threshold means that, while there are fewer protein-gene connections, the miRNA connections are maintained, thus the miRNAs influence is stronger.

One possible conclusion is that chaotic behaviours appear in networks with total inhibition rates between 50% and approximately 75%. Inhibition rates above 75% do not provide any dynamics while ordered dynamics appear preferably with low inhibition values up to 25% and cyclic dynamics appear, preferably, for inhibition rates from 25% to 50%.

It is also observed that ’max’ and ’random’ choices show little sensitivity to the binding threshold when compared to the other binding choices. This could mean that all networks created by ’max’ and ’random’ have approximately the same density of connections. Apparently these choices create networks too connected to produce interesting behaviours other than ordered dynamics. On the other hand the ’min’ choice also produces few dynamics of interest and, for this case, it may be that the network becomes too shallow with the increase of the binding threshold. The ’avg’ choice is the one showing more types of dynamics with a moderate density between ’min’ and ’max’ but still sensitive to the binding threshold.

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

The new model we propose, HeRoN, introduces a level of biological detail to previous models of gene regulatory networks. Separating the several processes and representing all the products involved in heterogeneous networks allowed, in particular, to extend the model to incorporate a RNA interference mechanism and make interesting observations about the topology and dynamics of the networks obtained.

From the static point of view, although many authors claim that genetic regulatory networks have scale-free topologies, most of them (Geard, 2004; Hallinan, 2004; Watson, 2004; Willadsen, 2003) are not based on experimental results for this concrete type of network but rather on other biological networks, such as the protein-protein interaction networks and metabolic networks. Furthermore, others (Liebovitch, 2006) use experimental mRNA concentration data to extract the networks ignoring all regulation other than regulation of transcription initiation. This could lead to misleading results since the presence of mRNA does not mean that the protein it codes for, a potential transcription factor, is actually synthesized, as other regulation mechanisms, such as the miRNA triggered inhibition, may be acting on it. A model that does not account for these mechanisms may incorrectly assume regulatory interactions between genes that are actually regulated by other products.

Another question of importance is that most models make a one-mode projection of an intrinsically heterogeneous network, i.e., they assume a network where all nodes are genes and the edges between them represent regulation relations. When such one-mode projection is made some information is obviously discarded (Newman, 2001). As was observed in real-world statistical data of other problems, real complex systems do not always have power law distributions because they are subject to constraints. In the HeRoN model we could not only find degree distributions that are constrained but we also introduced the study of degree distributions for some intermediate products.

Concerning the dynamics of the networks features such as stable and cyclic attractors, which exist in real biological networks, could be observed. The dynamics obtained seem to be closely related to the amount of negative connections with chaotic behaviors appearing for inhibition rates between 50% and 75%, cyclic behaviors appearing for 25% to 50% rates and ordered dynamics for less the 25% rates. The dynamic of a network seem also to be influenced by the density of connections as it could be observed that neither very dense nor very shallow networks would produce cyclic patterns.

This work, and the corresponding model, can be extended in several directions. Given the difference in dynamics shown between the smaller and bigger genomes, experiments with genomes of realistic dimensions should be performed. A good starting point would be the genome of the E.Coli since it is one of the smallest (4,600,000 bases long) and the most studied genome available. A case study of an actual biological system would then allow a more objective evaluation of the model. Moreover, the model could be improved and made sounder, by taking into account aspects such as the concentration of products (a continuous variable) and the time delays involved.

It would also be interesting to extend the model and observe how the alternative splicing of genes could alter the output degree distribution of genes, proteins and miRNAs. Finally, although several interesting observations were made by analyzing the degree distribution of the nodes, there are several other statistical properties that could be used to better understand them. Future work should include a study on the clustering coefficient, the average path length between nodes, the distribution of component (subgraph) sizes and the existence of a giant-component.

ACKNOWLEDGMENT

We thank Chris Stewart, Miguel Angel Rubio Escudero and Nicola Beveridge for their useful suggestions.

REFERENCES

Amaral, L., Scala, A., Barthélémy, M., & Stanley, H. (2000). Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America, 97*(21), 11149–11152. doi:10.1073/pnas.200327197

Ando, S., & Iba, H. (2001). Inference of gene regulatory model by genetic algorithms. In *Proceedings of the 2001 Congress on Evolutionary Computation, 1,* 712-719. IEEE-Press.

Baker, D., & Sali, A. (2001). Protein structure prediction and structural genomics. *Science, 294*(5540), 93–96. doi:10.1126/science.1065659

Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. In R. Riolo & B. Worzel (Eds.), *Genetic programming series, 6,* 43-62. Springer Verlag.

Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286,* 509–512. doi:10.1126/science.286.5439.509

D'haeseleer. P. (2000). *Reconstructing gene networks from large scale gene expression data*. Unpublished doctoral dissertation, University of New Mexico, USA.

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology, 91,* 67–103. doi:10.1089/10665270252833208

Geard, N. (2004). Modeling gene regulatory networks: Systems biology to complex systems. (Tech. Rep. ACCS Draft). Australia: The University of Queensland.

Goncalves, A., & Costa, E. (2007). *A computational model for genetic regulatory networks* (TR 2007/06, ISSN 0874-338X). Coimbra, Portugal: Universidade de Coimbra, CISUC.

Goncalves, A., & Costa, E. (2008). A computational model of gene regulatory networks and its topological properties. In S. Bullock, J. Noble, R. Watson & M. A. Bedau (Eds.), *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems* (pp. 204-211). Cambridge, MA: MIT Press.

Gregory, R., Chendrimada, T., & Shiekhattar, R. (2006). MicroRNA biogenesis: Isolation and characterization of the microprocessor complex. *Methods in Molecular Biology (Clifton, N.J.), 342,* 33–47.

Hallinan, J., & Wiles, J. (2004). Evolving genetic regulatory networks using an artificial genome. In Y. P. Chen (Ed.), *Proceedings of the Second Conference on Asia-Pacific Bioinformatics, 29,* 291-296. Darlinghurst, Australia: Australian Computer Society.

Hartl, D. L., & Jones, E. W. (2005). *Essential genetics: A genomic perspective*, 4th edition. Jones & Bartlett Publishers.

Hasty, J., & McMillen, D. (2001). Computational studies of gene regulatory networks. In numero molecular biology. *Nature Reviews. Genetics, 2,* 268–279. doi:10.1038/35066056

Hirata, H. (2002). Oscillatory expression of the bHLH factor Hes1 regulated by a negative feedback loop. *Science, 298*(5594), 840. doi:10.1126/science.1074560

Hoffman, M. M., Khrapov, M. A., Cox, J. C., Yao, J., Tong, L., & Ellington, A. D. (2004). AANT: The amino acid-nucleotide interaction database. *Nucleic Acids Research, 32*(Database issue), D174–D181. doi:10.1093/nar/gkh128

Kauffman, S. (1993). *The origins of order: Self-organization and selection in evolution.* USA: Oxford University Press.

Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., & Tomita, M. (2003). Dynamic modeling of genetic networks using genetic algorithm and s-system. *Bioinformatics (Oxford, England), 19*(5), 643–650. doi:10.1093/bioinformatics/btg027

Kuo, P., & Banzhaf, W. (n.d.). Small world and scale-free network topologies in an artificial regulatory network model. In J. Pollack, M. Bedau, P. Husbands, T. Ikegami & R.A. Watson (Eds.), *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (pp. 404-409).

Liebovitch, L., Jirsa, V., & Shehadeh, L. (2006). Structure of genetic regulatory networks: Evidence for scale free networks. In M. Novak (Ed.), *Complexus mundi: Emergent patterns in nature* (pp. 1-8). World Scientific.

Luscombe, N., Laskowski, R., & Thornton, J. (2001). Amino acid-base interactions: A three-dimensional analysis of protein-DNA interactions at an atomic level. *Nucleic Acids Research, 29*, 2860–2874. doi:10.1093/nar/29.13.2860

Martens, J., Laprade, L., & Winston, F. (2004). Intergenic transcription is required to repress the Saccharomyces cerevisiae SER3 gene. *Nature, 429*, 571–574. doi:10.1038/nature02538

Newman, M., Strogatz, S., & Watts, D. (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 64*, 026118. doi:10.1103/PhysRevE.64.026118

Noman, N., & Iba, H. (2005). Inference of gene regulatory networks using s-system and differential evolution. In H. Beyer (Ed.), *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation* (pp. 439-446). USA: ACM.

Reil, T. (1999) Dynamics of gene expression in an artificial genome-implications for biological and artificial ontogeny. In D. Floreano, J.-D. Nicoud & F. Mondada (Eds.), *Advances in Artificial Life: 5th European Conference* (pp. 457-466). Lausanne, Switzerland: Springer Verlag.

Reinitz, J., & Vaisnys, J. R. (1990). Theoretical and experimental analysis of the phage lambda genetic switch implies missing levels of cooperativity. *Journal of Theoretical Biology, 145*, 295–318. doi:10.1016/S0022-5193(05)80111-0

Sakamoto, E., & Iba, H. (2001). Inferring a system of differential equations for a gene regulatory network by using genetic programming. In *Proceedings of the 2001 Congress on Evolutionary Computation, 1*, 720-726. IEEE-Press.

Savageau, M. (1976). *Biochemical systems analysis: A study of function and design in molecular biology*. Reading: Addison-Wesley.

Thomas, R., & Kaufman, M. (2001). Multistationarity, the basis of cell differentiation and memory, II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos (Woodbury, N.Y.)*, *11*(1), 180–195. doi:10.1063/1.1349893

Vohradský, J. (2001). Neural network model of gene expression. *The FASEB Journal*, *15*, 846–854. doi:10.1096/fj.00-0361com

Voit, E. O., & Ferreira, A. (2000). Computational analysis of biochemical systems: A practical guide for biochemists and molecular biologists. Cambridge University Press.

Watson, J., Geard, N., & Wiles, J. (2004). Towards more biological mutation operators in gene regulation studies. [Elsevier.]. *Bio Systems*, *76*, 239–248. doi:10.1016/j.biosystems.2004.05.016

Willadsen, K., & Wiles, J. (2003). Dynamics of gene expression in an artificial genome. In *The 2003 Congress on Evolutionary Computation, 1*, 185-190. IEEE-Press.

Zhang, D., & Rosbash, M. (1999). Identification of eight proteins that cross-link to preRNA in the yeast commitment complex. *Genes & Development*, *13*(5), 581–592. doi:10.1101/gad.13.5.581

ADDITIONAL READING

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2008). *Molecular biology of the cell*, 3rd edition. Garland Science.

Alon, U. (2007). *An introduction to systems biology: Design principles of biological circuits*. Chapman and Hall.

Camazine, S., et al. (2001). *Self-organization in biological systems*. Princeton University Press.

Clote, P., & Backofen, R. (2000). *Computational molecular biology: An introduction*. Wiley & Sons.

Dividson, E. (2006). *The regulatory genome: Gene regulatory networks in development and evolution*. Academic Press.

Hallinan, J., & Wiles, J. (2004). Asynchronous dynamics of an artificial genetic regulatory network. In J. Pollack (Ed.), *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (pp. 404-409). USA: MIT Press.

Kuo, P., & Banzhaf, W. (2004). Small world and scale-free network topologies in an artificial regulatory network model. In J. Pollack (Ed.), *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (pp. 404-409). USA: MIT Press.

Kuo, P., Leier, A., & Banzhaf, W. (2004). Evolving dynamics in an artificial regulatory network model. In *Parallel problem solving from nature-PPSN VIII* (pp. 571-580). Springer Berlin/Heidelberg.

Miller, J., & Page, S. (2007). *Complex adaptive systems: An introduction to computational models of social life*. Princeton University Press.

Mitchell, M. (2006). Complex systems: Network thinking. *Artificial Intelligence, 170,* 1194–1212. doi:10.1016/j.artint.2006.10.002

Mitchell, M., Hraber, P., & Crutchfield, J. (1993). Revisiting the edge of chaos: Evolving cellular automata to perform computations. *Complex Systems, 7,* 89–130.

Watts, D. (1999). *Small Worlds: The dynamics of networks between order and randomness*. Princeton University Press.

KEY TERMS AND DEFINITIONS

Complex Systems: Said to be systems of interacting components with no central control that display emergent global properties not present at the components level.

Consensus Sequence: A nucleotide sequence composed of the most frequently observed base at each position among several observed sequences.

Degree Distributions: The probability distribution of degrees (number of edges between a particular node and the others) in a graph.

N - Partite: Graphs whose set of vertices is divided into n subsets, forming such a partition that no two vertices belonging to the same subset are adjacent.

One-Mode Projection: A one-mode projection of a n-partite network is the condensation of the representation of the network by representing and connecting nodes of only one type.

Skewed Distribution: A distribution is said to be skewed when one of its tails is longer than the other.

Unipartite: Graphs with only one type of vertexes as opposed to n-partite graphs.

Undirected: In an undirected graph both ends of an arc are equivalent.

Section 8

Other Studies

Chapter 23
Planning Interventions for Gene Regulatory Networks as Partially Observable Markov Decision Processes

Daniel Bryce
Utah State University, USA

Seungchan Kim
Arizona State University, USA

ABSTRACT

In this chapter, a computational formalism for modeling and reasoning about the control of biological processes is explored. It comprises five main sections: a survey of related work, a background on methods (including discussion of the Wnt5a gene regulatory network, the coefficient of determination method for deriving gene regulatory network models, and the partially observable Markov decision process model and its role in modeling intervention planning problems), a main section on the approach taken (including algorithms for solving the intervention planning problems and techniques for representing components of the problems), an empirical evaluation of the intervention planning algorithms on synthetic and the Wnt5a gene regulatory networks, and a conclusion and future directions section. The techniques described present a promising avenue of future research in reasoning algorithms for improved scalability in planning interventions in gene regulatory networks.

INTRODUCTION

Gene regulatory network (GRN) models, in their many forms (de Jong, 2002), provide a mathematical basis for representing and reasoning about biological processes. Of prime importance, is the use of GRNs to generate predictions of how a biological process changes over time through various forms of inference. One of the primary forms of inference is projection through simulation, especially in probabilistic GRN models. This chapter explores how to move beyond simulating biological processes, modeled as

DOI: 10.4018/978-1-60566-685-3.ch023

GRNs, and focuses on strategically inserting (planning) intervening actions to control the process. With connections to both shortest path problems and automated theorem proving, planning is a well-studied problem in Artificial Intelligence that focuses on generating sequences of actions (plans) that will transform an initial state into a goal state. The solution to the planning task (in this case an intervention plan) identifies how one might intervene with external control actions in abnormal development and prohibit cells from reaching undesirable states. This chapter explores planning interventions to avoid metastasis prone states in the WNT5A GRN.

The specific GRN model, described herein, represents a biological process by a set of genes and their regulatory influences over each other. The GRN focuses solely on genes (omitting proteins or other molecules) to model the high level behavior of many genes versus the low level behavior of a much smaller system. Because practical GRN models are typically learned from microarray data (Kim *et al*., 2000), they are situated at the right level of granularity for automated parameterization. As described below, microarray experiments measure the activity level of thousands of genes from living tissue in terms of mRNA concentrations (the products of gene transcription used to code proteins). Correlations between observed gene activity levels help describe regulatory influences. Predictor functions characterize the regulatory influences and provide a dynamic model (e.g., when genes g1 and g2 are highly active, gene g3 becomes inactive). The state of the GRN models the activity levels of genes and the predictor functions describe possible next states.

Layered on top of the GRN model, a planning model includes outside interventions (e.g., using RNA interference to suppress a gene's activity level) to alter the GRN predictor functions and effectively control the state evolution of the GRN. In an intervention plan, each action models either a possible intervention or non-intervention that will change the gene activity levels (i.e., the state of the GRN).

As one practical application of planning interventions, consider the treatment of cancer, where the chaining of multiple treatments is being aggressively explored. It is already clear that attention must be paid to the sequencing of these treatments. For example, cytotoxic drugs that induce replicative arrest and subsequent apoptosis (i.e. 5-fluorouracil or platinum containing drugs) rely on active replication to be effective. Cytostatic drugs (i.e. anti-estrogens, anti-angiogenics) on the other hand reduce ability to proliferate in order to reduce tumor load in patients and allow reductions in tumor size through active immunologic defense. Using a treatment sequence where a cytostatic drug is applied and still effective when the cytotoxic drug is given usually reduces the effectiveness of the cytotoxic drug, since far fewer tumor cells are actively replicating. The opposite sequence of treatment can be far more effective. It is likely that many treatment sequences will be either synergistic or antagonist based on timing and strengths of doses, and that a model that allows the physician to exploit the vulnerabilities of a disease to greatest therapeutic effect would be of great utility, not only for cancer treatment but also for other complex disorders.

While quite a bit of research remains to developing tools for designing treatment sequences, like those described above, this chapter discusses some of the relevant issues and a research direction that attempts to address the issues. Some of the important issues surrounding the design of treatment sequences (i.e., intervention plans) are (i) defining a suitable representation of the GRN model, (ii) building GRN models either manually or automatically from data, (ii) designing algorithms that reason with GRN models, and (iv) providing feedback to biologists on the nature and efficacy of computed solutions. As described below and in the Background Section, we represent the GRN model within a stochastic state transition system where transitions are controllable -- called a Markov decision process. While building GRN models is not the main emphasis of this chapter, we discuss how to learn a GRN model from microar-

ray data in the Background Section. The choice of algorithm used to reason with GRNs depends on the purpose; simulation algorithms described in the Background Section can be used to predict the steady state (stable) behavior of a GRN, and planning algorithms (the focus of this chapter, presented in the Planning Section) can be used to control the state changes of the GRN. Finally, providing feedback to biologists on solutions is important to validating their hypotheses about simulation or planning tasks; however, this chapter does not discuss this issue aside from pointing out useful metrics for comparing solutions in the section on Empirical Evaluation. We focus predominantly on the representation and reasoning algorithms necessary for constructing intervention plans because these are the most important new trends in systems biology, as evidenced by the increasing body of works that we discuss in the Related Work Section.

There are a number of planning formalisms that can capture GRN features; the formalism applied in this chapter is based on finite-horizon partially observable Markov decision processes (POMDPs), using the following motivations:

- Intervention plans need only focus on a number of steps (the horizon) long enough to ensure the GRN state will naturally transition to nominal states, and avoid abnormal states. Biological knowledge and computational simulation of GRNs (Kim *et al*., 2002) indicate that cellular processes, left to their own, transition to (or through) stable attractor states. These states represent common cellular phenomena such as the cell cycle, division, etc. However, some states are consistent with disease, such as the metastasis of cancer. From abnormal states, planning interventions provides a method to push the evolution of the process toward nominal attractor states.
- The cell has an inherently hidden state because full observations are prohibitively costly and inaccessible (Datta *et al*., 2004). Planning with partial observability is important because biologists analyzing cellular processes cannot be expected to understand, nor obtain complete state information.
- Biological processes are commonly viewed as stochastic (Elowitz *et al.*, 2002). Genes are typically regulated many different ways, meaning GRNs must allow for the probabilistic selection of predictor function for each gene.

Because state transitions are stochastic, provide only partial observations, and are only relevant over a limited planning horizon, the model for GRN interventions is based on decision theoretic planning (Boutilier *et al*., 1999), the problem of controlling a POMDP.

In addition to formulating the intervention planning problem as a POMDP, this chapter describes the application of a popular heuristic search algorithm used in AI planning, called *AO** (Nilsson, 1980) to efficiently solve the problem. This chapter illustrates the computational advantage of *AO** by showing how it avoids complete enumeration of all possible plans, and how it improves on existing dynamic programming techniques used for intervention planning (Datta *et al.*, 2004). An empirical evaluation on several variations of a synthetic GRN shows the benefit of heuristic search over these alternative techniques. An evaluation of planning interventions for the WNT5A GRN (Kim *et al*., 2002), identifies how benefits translate to a GRN of biological significance.

This chapter is organized as follows. A review of related work describes prior research on representing GRNs and planning interventions. A background section contains a significant amount of material on a motivating GRN (built around the WNT5A gene, which plays a role in the metastasis of melanoma), a method to construct a GRN from microarray data based on the coefficient of determination, and the

POMDP model and its role in representing GRN intervention planning problems. A section on the approach taken to generate intervention plans describes two algorithms (one based on heuristic search, called *AO**, and another based on dynamic programming) for solving problems represented as POMDPs and also discusses representation techniques used in conjunction with the algorithms. An empirical evaluation section compares both algorithms on several GRNs, either randomly generated or on the WNT5A GRN learned from actual microarray data. The chapter ends with a conclusion, discussion of future directions, an outlook, and list of supplemental reading.

RELATED WORK

Prior work related to the approach described in this chapter can be categorized into techniques for reasoning about and simulating GRNs as Boolean networks, approaches for controlling GRNs, and algorithms within Artificial Intelligence for reasoning about biological processes.

Boolean Networks as a Model for Gene Regulatory Networks

Studies of genetic regulatory networks can take many forms; some attempt to merely determine associative or predictive relations between genes, others seek to model network dynamics to the smallest biological nuance. A number of models have been proposed to study gene regulatory networks (de Jong, 2002). The complexity of these models ranges from simplistic models such as Boolean networks (Kaufmann, 1993, 1969), to more complete and intricate models based on differential equations (Goutsias & Kim, 2004, 2006). Evaluation of these models is done by examining how accurately the model reflects actual biology and how difficult the inference of a specific instance of the model is based on experimental data (de Jong, 2002). In general, as more low-level detail is added to a model, the more difficult it is to reconstruct the network from data.

A Boolean network is defined with a set of nodes (genes) and a set of Boolean functions. Each node acts as a binary variable to represent a gene's state at each time point, either expressed (1) or not expressed (0). At each time point *t*, the value of the node is updated based on the input of the genes at time *t-1* via a Boolean function. In practice, all of the nodes in a Boolean network are updated synchronously. One of the attractions to Boolean networks is that computationally efficient inference algorithms for Boolean networks have been presented (Lähdesmäki *et al.*, 2003).

Studies have shown that Boolean network models exhibit a number of biologically interesting properties (Kaufmann, 1969). Boolean networks primarily focus on determining gene-gene interactions at a qualitative level, instead of quantitative aspects. Additionally, Boolean networks can provide insight into cellular states. Both the steady-state and switch-like behavior of cells can be captured and studied with a Boolean model (Kim *et al.*, 2002). The ability to model both of these behaviors allows the analysis of common functions of the cell such as cell growth and cell cycle, as well as the response of a cell to external stimuli.

Beyond Boolean models, models exist that add a stochastic component. Models in this class include probabilistic Boolean networks (PBNs) (Shmulevich *et al.*, 2002a, 2002b), and Bayesian networks (Murphy and Mian, 1999; Friedman *et al*., 2000; Hartemink *et al.*, 2001). The addition of the stochastic component attempts to model both intrinsic and extrinsic noise in gene regulatory networks (Elowtiz *et al.*, 2002). In probabilistic Boolean networks, as in traditional Boolean networks, Boolean functions are

used to determine the next state of the network. Unlike traditional Boolean networks, however, there is not a single function corresponding to a network state. The next state of the network is determined via selection of Boolean function from a set of valid Boolean functions based on the current state. PBNs have been studied using Markov chains and shown to demonstrate both homeostasis and the switch-like properties exhibited by actual biological systems (Kim *et al*., 2002).

Controlling GRNs

Planning interventions in GRNs has been previously studied within the context of control theory, specifically, controlling Markov chains. AI planning and control theory have rich connections, and these similarities provide a common ground on which the empirical evaluations described in this chapter are based. The primary approach of previous work (Datta *et al*., 2003; 2004) on formulating intervention planning as a control problem was to characterize a dynamic programming operator that could identify the intervention to make in any state of the biological process. Dynamic programming uses the Bellman optimality principle to reduce the number of possible plans considered while finding the optimal plan. By construction, *optimal plans are composed of optimal plan suffixes*. That is, the second step of an optimal two step (horizon) plan must be an optimal one step plan. Since a multiple step plan can potentially visit any state during its execution, it becomes necessary to find all one step plans for all states, and then all two step plans for all states, and so on. Herein lies one of the problems with dynamic programming; even with knowledge of a start state, it does not necessarily limit those states considered for the plan and may generate many irrelevant subplans. In this formalism, Datta *et al.* (2003) formulate the problem as a fully observable Markov decision process and explore finite and infinite horizon control. Datta *et al.* (2004) also explore an extension to partial observability with finite horizon control. In both works, the underlying dynamics of the GRN is based on probabilistic Boolean networks and the techniques are evaluated within the WNT5A GRN. The difference, in terms of full or partial observability, is in how the plan respectively associates actions with states or probability distributions over states. This chapter describes partial observability, under the motivation that biological processes are for most practical purposes costly or difficult to fully observe in every detail.

Artificial Intelligence Approaches

Some recent works in the AI community have focused on simply representing biological processes. Khan *et al.* (2003) seek to discover signal transduction pathways with a deterministic classical planner. The actions in the plan represent various chemical reactions, and the goal of the plan is to establish that there is a sequence of actions leading to an event, such as transcription. This particular planning problem has received considerable attention due to its inclusion in a recent International Planning Competition. The most significant differences from the work described in this chapter, are that the model is assumed deterministic, is at a finer level of granularity (modeling many cellular products), and is primarily concerned with modeling the problem (versus exploring appropriate solution techniques).

Further along this vein of improved models of change in biological processes, Tran and Baral (2005) model change in biological processes as exogenous actions, termed *triggers*. In the planning considered in this chapter, the plans contain actions that represent both intervention and non-intervention, where the actions respectively model how the GRN changes under outside influence or naturally. Tran and Baral factor out the model of natural change in the biological process, representing it as triggers, which plans

can only indirectly affect. Actions become much simpler, in that they describe only the change due to the intervention, and not the coupling of the biological process and intervention.

BACKGROUND

This background section contains material on the WNT5A GRN and its significance. The section also describes the process of deriving a GRN from microarray data using the coefficient of determination (CoD), as described by Kim *et al.* (2000). With the derived GRN, the section continues by describing a POMDP based intervention planning formulation (Bryce and Kim, 2007). The following section describes how AI planning algorithms can be applied to efficiently solve this specific the POMDP formulation.

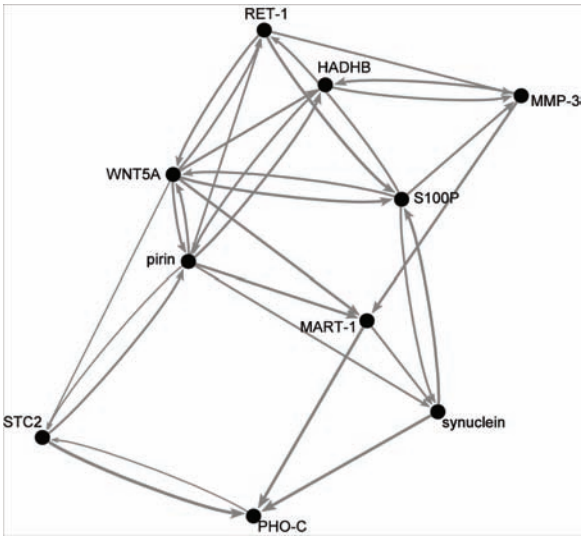
WNT5A Gene Regulatory Network

Wnt5a, a product of gene WNT5A, is a member of the Wnt family of proteins. The WNT gene family consists of structurally related genes which encode secreted signaling proteins. These proteins have been implicated in oncogenesis (malignant transformation leading to the formation of a tumor) and in several developmental processes, including regulation of cell fate and patterning during embryogenesis (embryo development) (Weeraratna *et al*., 2002). Specifically, the expression level of WNT5A is closely related with metastatic status (spread from one organ to another) of melanoma (Bittner *et al*., 2000). Later, it was also proved experimentally that increasing the level of Wnt5a protein can directly change the cell metastatic competence (Weeraratna *et al*., 2002; Sosman *et al*., 2004). It has been also suggested that controlling the influence of Wnt5a in the regulation can reduce the chance of melanoma metastasizing (Weeraratna *et al*., 2002).

In this chapter, we will focus on an in-silico gene regulatory network constructed via various previous studies (Bittner *et al*., 2000; Kim *et al.*, 2002; Weeraratna *et al*., 2002). The focus is the network of elements of the Wnt5a signaling pathway that deal with the phenotypic change from a less motile, less aggressive cell to a much more motile, more invasive cell. Early attempts at constructing mathematical models of the network of genes showing shared regulatory information with Wnt5a resulted in the identification of an interesting relationship between Wnt5a and Mart-1 (MLANA) expression (Figure 1, Kim *et al.*, 2002).

The network was constructed from a cDNA gene expression data containing probes for 8,150 cDNAs (representing 6,971 unique genes) and the 31 cutaneous melanoma tumor samples (Bittner *et al.*, 2000). To focus on a small set of important genes, prior work selected ten genes using the following criterion: (i) predictive relationships based on coefficient of determination (CoD) analysis (Kim *et al*., 2000; Dougherty *et al.*, 2000), (ii) roles in classifying malignant melanoma (Bittner *et al*., 2000), and (iii) biological functionalities. Specifically, prior work first identified a group of predictors that can simultaneously predict multiple target genes. The more target genes a set of predictive genes can predict well, the larger its extent of prediction is. Then, the authors also located genes that can be well predicted by many genes. Even though a causative relationship cannot be directly inferred from the coefficient of determination, the study was interested in a core group of genes that had strong cross predictivity, independent of the actual direction of action. Taking the intersection of these two gene sets both meets this requirement and reduces the number of candidates for the network. Further requirements for this core group of genes (alone or in combination) are that they should (i) show characterized biological functionalities; (ii) control and regulate the activity of other genes; (iii) modulate the phenotype of a cell.

Figure 1. Multivariate relationship between genes



Based on i) the coefficients of determination between each target gene and many possible predictors and ii) either their known or likely roles in the Wnt5a driven induction of an invasive phenotype in melanoma cells, ten genes were chosen. For these selected genes, the authors estimated CoDs of single-, two-, and three-gene predictors from the data. The highest CoDs for each target are shown in Table 1. Based on Table 1, we obtain the wiring diagram shown conceptually in Figure 1.

This network implied that high Wnt5a expression was linked with low Mart-1 expression. Mart-1 (MLANA) is a cell surface protein with epitopes (part of a macromolecule recognized by the immune system) commonly recognized by a class of tumor infiltrating lymphocytes (type of white blood cells) that have the ability to destroy melanoma cells (Kawakami *et al*., 1994; Cole *et al*., 1994). The coincidental alteration of melanoma cells to a more aggressive, mobile form, in conjunction with the damping of production of an antigen that can target melanoma cells for immunologic surveillance and killing

Table 1. The CoD values of the highest 3-to-1 combination for ten genes

Predictor 1	Predictor 2	Predictor 3	Target	CoD
WNT5A	STC2	HADHB	pirin	0.709
pirin	S100P	RET-1	WNT5A	0.683
WNT5A	RET-1	Synuclein	S100P	0.795
pirin	WNT5A	S100P	RET-1	0.625
S100P	RET-1	HADHB	MMP-3	0.700
MART-1	synuclein	STC2	PHO-C	0.920
pirin	WNT5A	MMP-3	MART-1	0.793
pirin	WNT5A	MMP-3	HADHB	0.772
pirin	S100P	MART-1	synuclein	0.559
pirin	WNT5A	PHO-C	STC2	0.479

makes this particular cascade interesting both in terms of cancer biology and therapy. The network has been extensively studied in the context of intervention (Datta *et al*., 2003; 2004).

The empirical evaluation section studies a seven gene version of this ten gene WNT5A network due to some representational challenges (discussed in detail below) that prevented consideration of the full network. The reduced network is also motivated by a need to compare with the previous work of Datta *et al.* (2004), where the authors used a seven gene network.

POMDP Model of GRNs

This subsection describes the finite horizon POMDP model, formally defines intervention planning, and details the formulation of intervention planning in the POMDP.

POMDP Model: The finite horizon POMDP problem *P* is defined as the tuple hS, A, T, R, Ω, O, h, b_Ii, where S is a set of states, A is a set of actions, T: S £A [{?}£ S[{?}![0,1] is a transition probability function, R: S £A [{?}£S[{?}! R is the transition reward function, Ω is a set of observations, O: S £A £Ω! [0, 1] is the observation function, h is the planning horizon, and b_I: S ![0, 1] is the initial belief state. We overload the symbol ? to denote both the terminal action and state signifying the end of the plan (i.e., the action and state at horizon h).

Consider a small example to illustrate the POMDP model. Let the set of states S be defined in terms of two states s1 and s2, such that S = {s1, s2}. Let the set of actions A = {I1, NI}, where "I1" signifies intervening by inhibiting a gene g1 and "NI" signifies not intervening. Let the transition probability function T and transition reward function R be defined as follows:

T(s1, I1, s1) = 1.0 R(s1, I1, s1) = -1

T(s1, I1, s2) = 0.0 R(s1, I1, s2) = -1

T(s2, I1, s1) = 0.8 R(s2, I1, s1) = -1

T(s2, I1, s2) = 0.2 R(s2, I1, s2) = -1

T(s1, NI, s1) =.5 R(s1, NI, s1) = 0

T(s1, NI, s2) =.5 R(s1, NI, s2) = 0

T(s2, NI, s1) =.25 R(s2, NI, s1) = 0

T(s2, NI, s2) =.75 R(s2, NI, s2) = 0

T(s1, ?, s1) = 1.0 R(s1, ?, s1) = 0

T(s2, ?, s2) = 1.0 R(s2, ?, s2) = -10

Let the set of observations contain two observations o1 and o2, such that Ω = {o1, o2}, and let the observation function O be defined for every action as:

$O(s1, \cdot, o1) = 0.7$ $O(s1, \cdot, o2) = 0.3$

$O(s2, \cdot, o1) = 0.3$ $O(s2, \cdot, o2) = 0.7$

Let the initial belief state b_I be defined as $b_I(s1) = 0.5$ and $b_I(s2) = 0.5$.

In this POMDP, the reward function indicates that the state s1 is a desirable end state because self transitions (those after the number of steps in the horizon) due to the ? action are rewarded 0, whereas the same action for s2 rewards -10. The reward for the terminal action can only be accrued once at the end of the plan. The intervention action I1 leads to s1 with high probability from both states and has a small cost (negative reward). The non-intervention action NI has no cost and, depending on the current state, transitions to the other with some probability (e.g., in s1 NI transitions to s2 with 0.5 probability). The observation function states that receiving observation o1 indicates that the current state is s1 with 0.7 probability, or that it is s2 with 0.3 probability. The initial belief state assumes that either state is the initial state with 0.5 probability. As the next section defines, it is possible to compute the belief states reached by executing each possible action at the first time step and receiving an observation, using Bayes rule. Repeated computation of belief states reachable by different action and observation sequences (up to the horizon h) provides the information sufficient to find a solution, called a conditional plan (i.e., the best actions to perform for each observation sequence). However, full computation of every sequence is not always necessary, as we will see in the next section.

Intervention Planning Problem: Given our definition of the POMDP model, the following shows how an intervention planning problem maps to the POMDP. The intervention problem is defined by the tuple h*G*, Dom, F, X, W, Y, *O*, hi, where *G* is a set of genes, Dom is the set of activity levels for genes, F is a set of predictor functions, X is a set of interventions, W is an initial situation, Y is a goal description, *O* is a set of observations, and h is the horizon. The genes and their activity levels describe states of the POMDP, the predictor functions and interventions describe actions, interventions and the goal description define the reward function, and the initial situation, observations, and horizon map directly to their POMDP counterparts.

States: Each gene g 2*G* has an activity level from the domain Dom of values, of which we will only illustrate Boolean domains {g,:g}, active or inactive. A state s: *G!*Dom of the gene network maps each gene to a value d2Dom. The entire set of GRN states defines the POMDP states S.

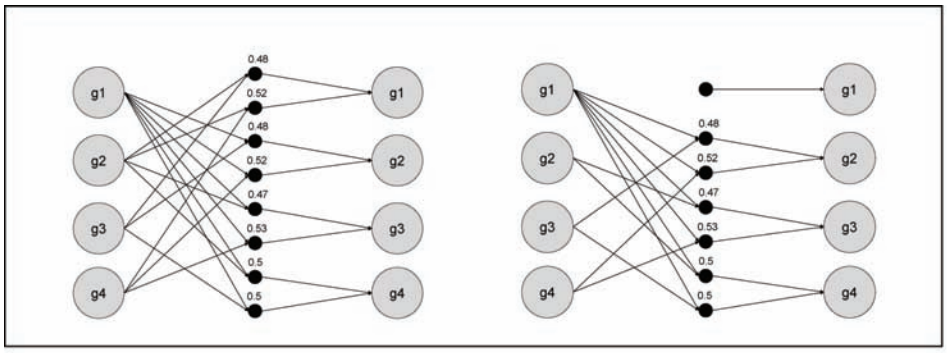
For example, a four gene problem G = {g1, g2, g3, g4} would give rise to a sixteen state POMDP with states S = {s1 = {g1, g2, g3, g4}, s2 = {:g1, g2, g3, g4}, …, s16 = {:g1,:g2,:g3,:g4}}

Predictor Functions and Interventions: Given a state of the gene network, the predictor functions F are used to describe states reachable after one step. Interventions re-write predictor functions in F for specific genes to ensure the gene network transitions to specific states. Thus, each possible action in the POMDP is described by a set of predictor functions. A non intervention simply uses F to describe the action, but an intervention action x 2X replaces predictor functions in F to get a new set F_x. Each intervention x 2X is a set of predictors $\{ f_{g_1}, f_{g_2}, \ldots \}$, allowing us to define

$F_x = F[x\{ f_g \mid f_g \in F, f_{g'} \in x, g = g'\}$.

Each predictor function f_g is defined as the mapping $f_g : \text{Dom}^{|G'|} \to \text{Dom}$ from activity levels of genes in $G' \mu G$ to the activity level of gene g. The interventions described in this chapter contain a single

Figure 2. Graphical depiction of predictor functions for a non-intervention (left) and intervention (right) action



predictor function f_g where $G'=\emptyset$, meaning that, irrespective of the state, g has its activity level set deterministically.

Since each gene may be predicted by several predictor functions, where each state transition selects one probabilistically, each predictor is assigned a weight w(f_g) (based on its normalized CoD). The empirical analysis evaluates GRNs where the predictor functions and weights are both generated randomly and from real microarray experiments.

Figure 2 depicts two actions in a four gene GRN, a non intervention action on the left and an action intervening g1 on the right. The genes on the left side of each action symbolize current state values for the genes, and the left symbolize the next state values. The dark circles indicate different predictor functions with their associated weights. Each of the predictor functions uses two genes to determine the next value of a gene, and there are two predictor functions per gene. For example, the non-intervention action has two predictors for g1, f_{g1}(g2, g3) and f_{g1}(g2, g4), with the respective weights 0.48 and 0.52. The first predictor function for g1 might be defined as follows:

g2, g3) =:g1

f_{g1}(g2,:g3) = g1

f_{g1}(:g2, g3) =:g1

f_{g1}(:g2,:g3) = g1

to indicate the activity levels of g2 and g3 that map to various activity levels for g1 in the next state. The intervention action uses a single predictor for gene g1, effectively overwriting the predictors in the non-intervention action and leaving all other predictor functions as before.

Each action a_F defined by the set of predictor functions F (similarly F_x) describes the transition probability function

Figure 3. Transition probability matrix for the intervention action depicted in Figure 2

			g1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
			g2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			g3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
g1	g2	g3	g4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1.00	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0.23	0.00	0.25	0	0.24	0	0.28	0
0	0	1	0	0	0	0	0	0	0	0	0	0.26	0.26	0	0	0.24	0.24	0.00	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0.24	0.24	0.27	0.27
0	1	0	0	0	0	0	0	0	0	0	0	0.27	0.27	0.24	0.24	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0.24	0.24	0	0	0.26	0.26
0	1	1	0	0	0	0	0	0	0	0	0	0	0.28	0	0.24	0	0.25	0	0.23
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.47	0	0.53
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.50	0.50
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0.24	0.24	0.27	0.27
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.00
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.53	0	0.47
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.50	0.50
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0.27	0.27	0.24	0.24

$$T(s, a_F, s') = Pr(s' \mid F, s) = \prod_{g \in G} \left(\frac{\sum_{f_g(s) \in F : f_g(s) = s'(g)} w(f_g(s))}{\sum_{f_g(s) \in F} w(f_g(s))} \right).$$

The matrix in Figure 3 is one representation of the transition probability function for the intervention action on the right in Figure 2. The rows correspond to the current state s and the columns refer to the next state s', and the activity level for each gene is shown for each state. Notice that because the intervention action ensures that the activity level for g1 is active in each next state, the transition probability matrix has all zero entries for any next state where g1 is inactive (highlighted grey entries). However, there are still a considerable number of zero entries in the matrix due to state transitions not possible in the GRN model. In the next section, an alternative, more compact, representation of the transition probability matrix is described.

Observations: After each step, whether by intervention or non intervention, the state of the gene network is observed. The set O µ G defines which genes are observable (by genetic markers, physiology, etc.). The set of observations $\Omega = \{o \mid o\ 2Dom^{|O|}\}$ is defined by all joint activity levels of genes in O. This chapter assumes that observations are perfect and the same for each action, meaning that if a state s and observation o agree on the activity level of each gene, then the probability of the observation is one (i.e., O(s, a, o) = 1, otherwise zero (i.e., O(s, a, o) = 0). Observations can be noisy (i.e., 0 ·O(s, a, o) ·1) in general.

For example, the four gene GRN might have the observables O = {g1, g3}, making the set of possible observations Ω = {o1 = {g1, g3}, o2 = {:g1, g3}, o3 = {g1,:g3}, o4 = {:g1,:g3}}. An observation function that models perfect observations would have the following example values:

O({g1, g2, g3, g4}, a, o1) = 1

O({:g1, g2, g3, g4}, a, o1) = 0

O({g1,:g2, g3, g4}, a, o1) = 1

because in the first case o1 matches the values of genes in the state, in the second case the state does not match the observation, and in the third case the state matches the observation. The observation function can be defined similarly for all states and observations.

Rewards: The goal Y is a function describing desirable states. The goal maps states to real values Y:Dom$^{|G|}$!R. The reward function for terminal actions and goal states is defined by the goal R(s, ?, ?) = Y(s). This chapter assumes that the reward associated with actions is -1 for intervention actions (i.e., R(s, a_{F_x} , s') = -1) and 0 for non intervention (i.e., R(s, a_F, s') = 0).

Initial Situation: The initial situation W is a distribution over GRN states W: Dom$^{|G|}$![0, 1]. This mapping to the POMDP initial situation is straight-forward, b_1 (s)=W(s).

Since the formulation has a distinct initial belief state, the approach explored next used heuristic search and knowledge of the initial belief state to guide expansion of a conditional plan. By searching forward from the initial belief state, it is possible to focus plan construction on reachable belief states.

PLANNING

This section describes an approach to solving the finite horizon POMDP representing the GRN intervention planning problem. First, the semantics of conditional plans and the search space are defined; these are followed by a discussion of action and belief state representation, along with two search algorithms to find plans. The first algorithm is *AO** (Nilsson, 1980), and the second *Datta* is based on a competing approach (Datta *et al*., 2004) from the GRN literature.

Conditional Plans

A solution to the problem *P* is a conditional plan P of horizon h, described by a partial function P: V !A [{?} over the belief state space graph G = (V, E). A subset of the vertices b 2V (which are belief states) are mapped to a "best" action a, denoted P(b) = a. Each edge e2 E directed from b to b_a^o is mapped to an action a 2A and an observation o 2 Ω, and denoted e(b, b_a^o) = (a, o). If P(b) = a and e(b, b_a^o) = (a, o), then P(b_a^o) is the action to execute after executing a and receiving observation o. Throughout the discussion, it is assumed that the horizon is a feature of every state to ensure that the graph G is acyclic. Belief states where the horizon is equal to h have a single available action ? to signify the end of the plan, leading to a terminal ?.

If P(b) = a, and there exists an edge e(b, b_a^o) = (a, o) then the successor belief state b_a^o is defined

$$b_a^o(s') = \alpha b_a(s')O(s', a, o),$$

where

$$b_a(s') = \sum_{s \in S} b(s)T(s, a, s'),$$

and ® is a normalization constant. If for all s2 S, $b_a^o(s) = 0$ because no observation is consistent with the belief state b_a, then the belief state is not added to the graph. For example, in the two state POMDP from the previous section, applying action I1 in the initial belief state b_I results in b_{I1} (s1) = b_I (s1)T(s1, I1, s1) + b_I (s2)T(s2, I1, s1) = 0.5(1.0)+0.5(0.8) = 0.9 and b_{I1} (s2) = b_I (s1)T(s1, I1, s2) + b_I (s2)T(s2, I1, s2) = 0.5(0.0)+0.5(0.2) = 0.1. With the observations we have:

$b_{I1}^{o1}(s1) = \alpha b_{I1}$ (s1)O(s1, I1, o1) = $\left(\frac{1}{0.9(0.7)+0.1(0.3)}\right)$ 0.9(0.7) = 0.95,

$b_{I1}^{o1}(s2) = \alpha b_{I1}$ (s2)O(s2, I1, o1) = $\left(\frac{1}{0.9(0.7)+0.1(0.3)}\right)$ 0.1(0.3) = 0.05,

$b_{I1}^{o2}(s1) = \alpha b_{I1}$ (s1)O(s1, I1, o2) = $\left(\frac{1}{0.9(0.3)+0.1(0.7)}\right)$ 0.9(0.3) = 0.79, and

$b_{I1}^{o2}(s2) = \alpha b_{I1}$ (s2)O(s2, I1, o2) = $\left(\frac{1}{0.9(0.3)+0.1(0.7)}\right)$ 0.1(0.7) = 0.21

and similarly we find that $b_{NI}^{o1}(s1)$ = 0.53, $b_{NI}^{o1}(s2)$ = 0.47, $b_{NI}^{o2}(s1)$ = 0.18, and $b_{NI}^{o2}(s2)$ = 0.82.

The expected reward q(a, b) of a plan that starts with action a at belief state b is the sum of current and future rewards:

$$q(a,b) = \sum_{s\in S} b(s) \sum_{s'\in S} T(s,a,s')(R(s,a,s') + \sum_{o\in\Omega} b_a(s')O(s',a,o)V(b_a^o)),$$

where the expected reward for a belief state is *V*(b). Terminal vertices are assigned the expected goal reward

$$q(\perp,b) = \sum_{s\in S} b(s)R(s,\perp,\perp).$$

In the example, if we assume that *V*(b_{I1}^{o1}) = -7.4, *V*(b_{I1}^{o2}) = -3.2, *V*(b_{NI}^{o1}) = -9.1 and *V*(b_{NI}^{o2}) = -10.3, then we have:

q(I1, b_I) = b_I(s1)(T(s1, I1, s1)(R(s1,I1,s1) + b_{I1} (s1)O(s1, I1, o1) *V*(b_{I1}^{o1})) +

T(s1, I1, s2)(R(s1,I1,s2) + b_{I1} (s2)O(s2, I1, o1) *V*(b_{I1}^{o1}))) +

b_I(s2)(T(s2, I1, s1)(R(s2,I1,s1) + b_{I1} (s1)O(s1, I1, o2) *V*(b_{I1}^{o2})) +

T(s2, I1, s2)(R(s2,I1,s2) + b_{I1} (s2)O(s2, I1, o2) *V*(b_{I1}^{o2})))

= 0.5(1.0 (-1.0 + 0.9(0.7)(-7.4)) +

0.0 (-1.0 + 0.1(0.3)(-7.4))) +

0.5(0.8 (-1.0 + 0.9(0.3)(-3.2)) +

0.2 (-1.0 + 0.1(0.7)(-3.2))) = 3.7

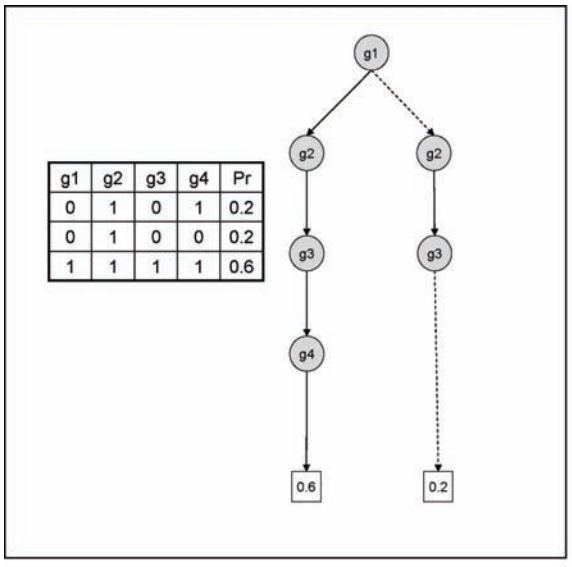
The values V(b) can be computed in many ways, below we describe an iteratively converging approach in the *AO** algorithm and a dynamic programming algorithm studied by prior work. The actions that define the optimal values correspond to the actions defining an optimal plan.

Representation

Actions and belief states represent probability distributions over state transitions and states, respectively. A practical consideration of any algorithm that constructs the belief state space graph is how to compactly represent a large number of probability distributions. It is common in AI planning to make use of the factored representation of states to capture structure compactly. In this case, state factors are genes.

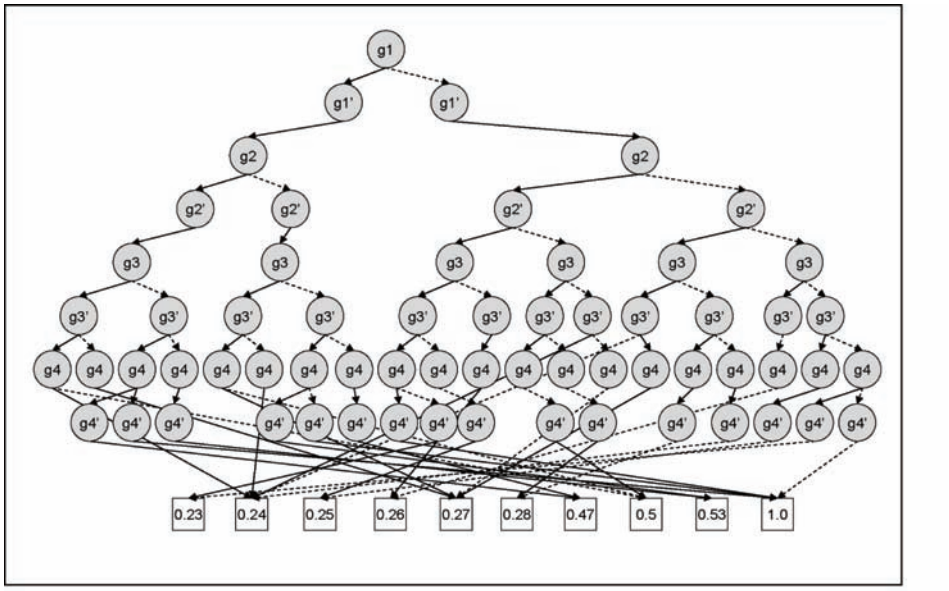
Consider the table and diagram in Figure 4, called an algebraic decision diagram (ADD) (Bryant, 1986). Each row in the table corresponds to a state, where the activity level for each gene is given, and the probability of the state in the belief state is given. The first two states have equal probability, and differ only in the last gene g4; an ADD can exploit this structure. The ADD is a graph representing state factors (genes) as nodes, and their values (activity levels) as edges. The solid edges indicate the value is "1" and the dashed edges indicate the value is "0". Each path through the ADD from the root to a leaf node corresponds to a set of potential states. The path on the right corresponds to the first two states;

Figure 4. Belief state and ADD representation



g1	g2	g3	g4	Pr
0	1	0	1	0.2
0	1	0	0	0.2
1	1	1	1	0.6

Figure 5. ADD representing state transition probability for the action intervening gene g1



because the node for g4 does not appear on the path, its value adds no information in representing these two states in the probability distribution. In this fashion it is possible to represent a large number of states very compactly when the states have identical probabilities within a belief state.

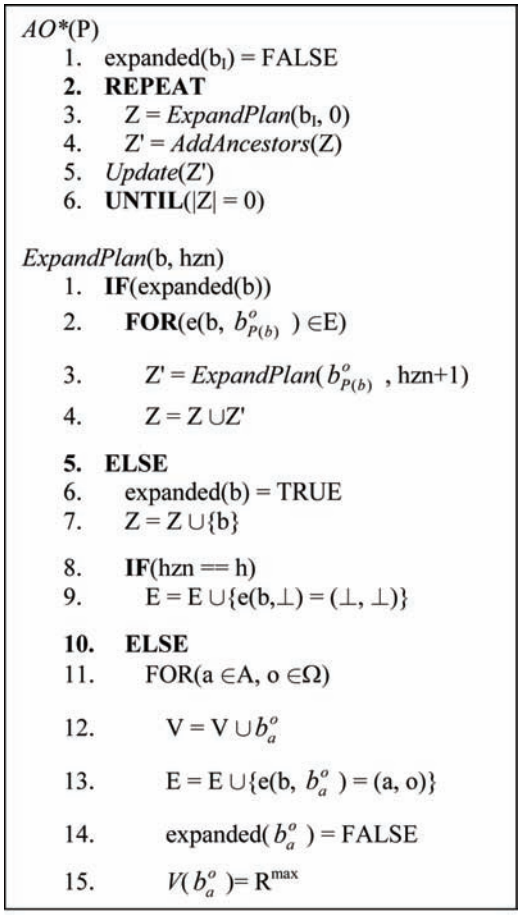
Similarly, it is possible to represent the state transition probability distribution for each action using an ADD. Figure 5 depicts the ADD for the intervention action described in the previous section. The state transition probability must represent a probability for each pair of states, meaning that the ADD must use nodes to represent both the predecessor state factors and the successor state factors (those nodes in Figure 5 whose factor has a dash). In the same spirit as the ADD for the belief state, each path through the ADD in Figure 5 represents a set of state transitions and their probability. Notice that because the action intervenes to activate g1, all paths through g1' use a solid edge, meaning that g1' must be active in all successor states. Action observation probability functions are represented in a similar fashion.

It is possible to use efficient (polynomial time) ADD algorithms to compute the successor belief states (for all constituent states at once) in the formulas given in the discussion of conditional plans above. The main operation involves a product between the belief state ADD with the transition ADD, followed by summation over predecessor states in the resulting ADD. These algorithms are beyond the scope of this chapter, and the interested reader is referred to (Meinel and Theobald, 1998).

*AO** Algorithm

It is possible to solve the finite horizon POMDP problem with *AO** search (Nilsson, 1980) in the space of belief states. The *AO** algorithm, listed in Figure 6, takes the planning problem as input, and iteratively constructs the belief space graph G rooted at b_I. The algorithm involves three iterated steps: expand the current plan with the *ExpandPlan* routine (line 3), collect the ancestors Z' of new vertices Z (line 4), and compute the current best partial plan (line 5). The algorithm ends when it expands no new vertices. The following briefly describes the sub-routines used by *AO**.

Figure 6. AO search algorithm*



```
AO*(P)
1.  expanded(b_I) = FALSE
2.  REPEAT
3.    Z = ExpandPlan(b_I, 0)
4.    Z' = AddAncestors(Z)
5.  Update(Z')
6.  UNTIL(|Z| = 0)

ExpandPlan(b, hzn)
1.  IF(expanded(b))
2.    FOR(e(b, b^o_P(b) ) ∈E)
3.      Z' = ExpandPlan( b^o_P(b) , hzn+1)
4.      Z = Z ∪Z'
5.  ELSE
6.    expanded(b) = TRUE
7.    Z = Z ∪{b}
8.    IF(hzn == h)
9.      E = E ∪{e(b,⊥) = (⊥, ⊥)}
10. ELSE
11.   FOR(a ∈A, o ∈Ω)
12.     V = V ∪ b^o_a
13.     E = E ∪{e(b, b^o_a ) = (a, o)}
14.     expanded( b^o_a ) = FALSE
15.     V( b^o_a )= R^max
```

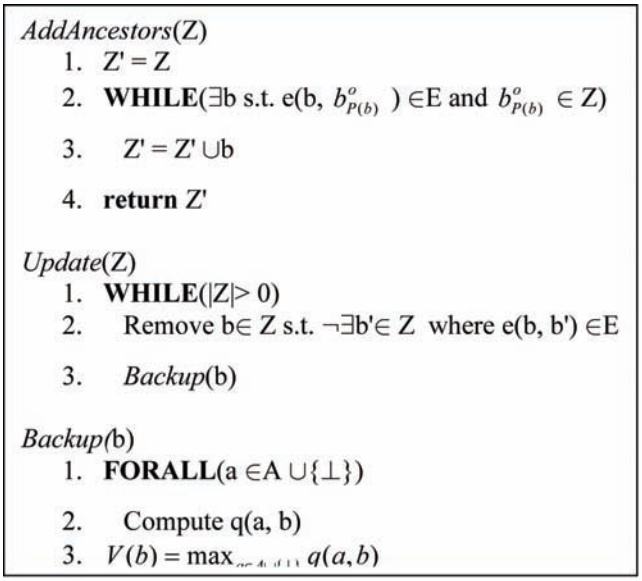
The *ExpandPlan* routine recursively walks the current plan to find unexpanded vertices (lines 2-4). Upon finding a vertex to expand, it generates all successors of the vertex (lines 5-15). Generating successors involves assigning the ? action if the vertex is at the max horizon (line 9) or constructing the vertices reached by all action and observation combinations (lines 11-15). Notice that each vertex has its value initialized with an upper bound,

$$R^{\max} = \max_{\substack{s,s' \in S \\ a \in A}} R(s,a,s')h + \max(0, R(s,\perp,\perp)),$$

on its expected reward. The upper bound plays a role in pruning vertices from consideration in the search.

After expanding the current plan, *ExpandPlan* returns the set of expanded vertices Z. In order for *Update* (Figure 7) to find the best plan, given the new vertices, *AddAncestors* adds to Z' every ancestor vertex of a vertex in Z. The resulting set of vertices consists of every vertex whose value (and best action) can change after *Update*. The *Update* routine iteratively removes vertices from Z that have no descendent in Z and calls *Backup* until no vertices remain in Z. The *Backup* routine computes the value

Figure 7. AO subroutines*

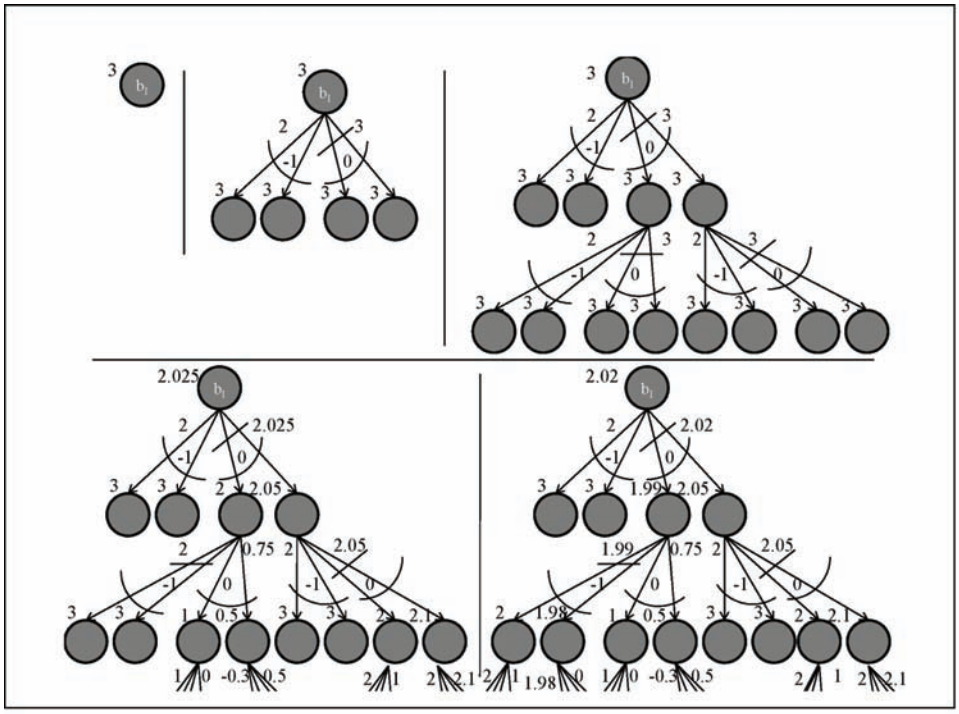


of a vertex and sets its best action. The reason *Update* chooses vertices with no descendent in Z is to ensure each vertex has its value updated with the updated values of its children.

*AO** can often avoid computing the entire belief space graph G, leading to significant savings in problems with large horizons. By initializing vertices with an upper bound on their value it is possible to ignore vertices that have a consistently lowest upper bound. For example in *Backup*, if there exists an action whose q-value is always greater than the alternative actions, then the best action will never be set to one of the alternatives. Further, because the alternative actions are never considered best, *ExpandSolution* will never expand them. As explored in the empirical evaluation, the reward function has a significant effect on the number of vertex expansions. In the worst case, it is possible to expand the entire graph G, as would the *Datta* algorithm.

Consider the example shown in Figure 8 of *AO** expanding the belief state space to find a conditional plan. The belief state space graph begins with a single node for the initial belief state (top left of the figure), which is given an initial value $V(b_l) = R^{max} = 3$. In the first call to *ExpandSolution*, four successor belief states are generated, one for each action and observation pair. The edges are grouped by with a curved arc to indicate that they correspond to alternative observations but the same action. The number in between the connected edges indicates the immediate reward of the action, either "-1" to intervene or "0" to not intervene and let the GRN change on its own. Each new belief state is assigned a value $R^{max} = 3$. *AO** then updates the values of the actions to "2" and "3" (shown on the outside of the edges), and the value of b_l. The best action to perform in b_l maximizes its value, meaning that non-intervention is best, and the straight line connecting the edges for that action indicate that it is best and part of the current best partial plan. At the next iteration, *ExpandSolution* generates the successor belief states for the belief states reached by following the current best partial plan, and the values are updated as before. The next iteration (lower left of the figure) again generates the successors of belief state in the current plan. Assuming the planning horizon is three, the newly generated successors are assigned values to indicate the expected reward received by satisfying the goal (whose value is 3). The value updating identifies a

Figure 8. AO example*



new best action in the left-most branch of the plan, marking the intervention action, but in the right-most branch the non-intervention action is best. While *AO** expanded the plan up to the maximum horizon, the updating changed the plan, requiring additional expansion. That is, *Update* propagates refined values back to the root node; because values may be only estimates in some cases, refined values can change the best partial plan and require additional exploration of the search graph. In the last iteration, *AO** expands again and the values change, but no new plan is identified, so it terminates with a plan of value 2.02. Notice that plans starting with the first intervention action are never explored; the upper bound on its value is two, and all partial plans explored by the algorithm had a value greater than two, meaning they were always better. Using even a loose upper bound can avoid generating unnecessary regions of the belief state space, and having tighter upper bounds can only improve the savings. However, computing such tighter bounds can be difficult and are an area of intense research in AI.

Datta Algorithm

In order to compare the *AO** planner to the work of Datta *et al.* (2004), Figure 9 provides a description of their algorithm, which will be referred to as *Datta*. Unlike the iterative *AO**, *Datta* consists of two steps: expand G with *ExpandPlanD*, and then update each vertex v2V with *Update*. The *ExpandPlanD* routine recursively expands G by either reaching a terminal vertex at the horizon (line 2), or generating and recursing on each child of a vertex (lines 5-7). Following *ExpandPlanD*, Update computes the best action for each vertex in V.

Unlike *AO**, *Datta* is unable to prune vertices from expansion, making it insensitive to the reward function. While the result of the two algorithms is identical, the time and space required can be very

different. Both algorithms were implemented within the same planner and the empirical evaluation demonstrates their effectiveness on the GRN intervention planning problems. The authors also implemented a straight-forward version of the Datta *et al*. (2004) algorithm that does not make use of several efficiency improvements within the planner, such as using ADDs Bryant (1986) for compact action and belief state representation, as well as duplicate belief state detection.

Empirical Evaluation

This section describes an empirical evaluation aimed to not only explore the limits of the planning techniques (based on *AO** search and the ADD representation) described for intervention planning, but also compare with exiting algorithms (namely the *Datta* algorithm). The evaluation is broken into three parts. The first describes the test setup, and the following two describe a comparison of the algorithms on two GRNs and three intervention problems, and an internal evaluation to explore the scalability of the described approach.

Setup: The experiments, described below, are designed to test the feasibility of using AI planning techniques to solve GRN intervention problems, and are based on several GRNs, both randomly generated and the learned WNT5A GRN (Kim *et al*., 2002). The random variations are meant to stress different aspects of the planner, including its ability to prune the belief state space using bounds and the limits of its representation of the GRN and intervention actions as ADDs. Table 2 summarizes the features of the GRNs. All GRNs use two predictor functions per gene, each with two genes as predictors, as well as one observable gene. The first two GRNs help to illustrate the differences between the *AO** and *Datta* algorithms; the random GRN "Random7-3" has seven genes and three intervention actions, and the WNT5A GRN has seven genes and one intervention action (two different intervention actions are used below). The WNT5A GRN is formulated to match the GRN presented by Datta *et al.* (2004), and the Random7-3 GRN is meant to explore how the algorithms behave when the number of interventions are increased. The predictor functions are selected randomly in the random GRNs and learned from microarray data (using the coefficient of determination method described above) in the WNT5A GRN. (In the WNT5A GRN, the two predictor functions for each gene have the highest CoD, effectively setting their

Figure 9. Datta algorithm

```
Datta(P)
    1. ExpandPlanD(b_I, 0)
    2. Update(V)

ExpandPlanD(b, hzn)
    1. IF(hzn == h)
    2.    E = E ∪{e(b,⊥) = (⊥,⊥)}
    3. ELSE
    4.    FOR(a ∈A, o ∈Ω)
    5.       V = V ∪{ b_a^o }
    6.       E = E ∪{e(b, b_a^o ) = (a, o)}
```

Table 2. Test GRNs

	|G|	|Dom|	|F|	|X|	|O|
Random7-3	**7**	**2**	**14**	**3**	**1**
WNT5A	**7**	**2**	**14**	**1**	**1**
Random7-1	**7**	**2**	**14**	**1**	**1**
Random8-1	**8**	**2**	**14**	**1**	**1**
Random9-1	**9**	**2**	**14**	**1**	**1**

weight proportional to the CoD.) The last three GRNs "Random7-1", "Random8-1", and "Random9-1" increase the number of genes in the GRN from seven to nine and use a single intervention, as indicated by their names. While the WNT5A network described in the Background section contained ten genes, planning interventions is a more computationally difficult problem, relegating us to small GRNs, of at most nine genes, with Boolean gene activity levels. (The future research directions section, below, indicates techniques that may be employed to increase the number of genes.)

There are several intervention problems studied in the GRNs. In the Random7-3 GRN, the goal W is varied to assign different rewards to terminal states, while assigning interventions a negative reward of one. This illustrates the ability of *AO** to prune the search space in comparison with enumeration. The WNT5A GRN allows the reproduction of two intervention problems studied by Datta *et al*. (2004). The first directly intervenes to suppress WNT5A (which happens to be the goal) and observes the pirin gene. The second attempts to indirectly control WNT5A by pirin (a predictor gene of WNT5A) intervention. Both WNT5A problems use the same reward function, assigning interventions a negative reward of one and the goal (activating WNT5A) a negative reward of three. The goal accrues negative reward to maintain consistency with the Datta *et al*. (2004) model. Thus plans avoid activating WNT5A, which is equivalent to deactivation. Both WNT5A networks use the initial belief state where each gene is set to an activity level with probability proportional to its observed frequency in the data. The last three networks are used to explore the internal scalability of the AI planning approach by increasing the number of genes, and holding constant the number of interventions at one intervention. Each problem assigns a reward of ten to the goal. A common factor scaled across all intervention planning problems is the horizon of the plan. As the horizon increases, the number of possible plans and the number of decision points grow exponentially, making the problem more challenging.

The planner is implemented in C++ and ran on a machine with 1GB of RAM. The experiments involving the first two GRNs were given a twenty minute time limit and the experiments on the last three were given thirty minutes (to allow for long preprocessing time, converting the GRN into ADDs).

Algorithm Comparison

The leftmost plots in Figure 10 depict the number of expanded vertices (including terminal actions) in *AO**, *Datta*, and the maximum possible (Max), and the rightmost plots depict the total run time in seconds for the corresponding plot on the left. The results for *AO** are indexed by a number indicating the reward associated with the goal, since *Datta* is insensitive to reward. Max represents the number of vertices expanded in a search tree (versus a graph), similar to the original implementation of Datta *et al*. (2004). By implementing the *Datta* algorithm within the planner, like *AO** it finds duplicate belief states. Without duplicate detection, *Datta* would expand as many vertices as Max. The leftmost plot in

Figure 10 shows the associated total planning time (which is proportional to the number of expanded vertices). Missing bars indicate the instance was cut off by reaching the 20 minute time limit.

There are several important points to note in the results. *AO** is sensitive to the goal reward function, expanding much fewer vertices than *Datta* in some cases. Despite *AO** using dynamic programming over its partial solution many times, it never takes more time than *Datta*. When it is able to prune vertices, *AO** scales much better.

The second row of plots in Figure 10 show results in the WNT5A GRN intervening WNT5A and observing pirin. Here *AO** greatly outperforms *Datta*. The difference is partly due to the fact that *AO** quickly recognizes that the optimal plan intervenes once at the end of each plan branch where WNT5A is not already inactivate. A direct implementation of the approach of Datta *et al*. (2004), which does no duplicate detection was able to solve this problem to a horizon of ten, using significantly more memory

Figure 10. Number of expanded vertices (left) and total planning time in seconds (right) for random GRN (top), WNT5A intervention (center), PIRIN intervention (bottom).

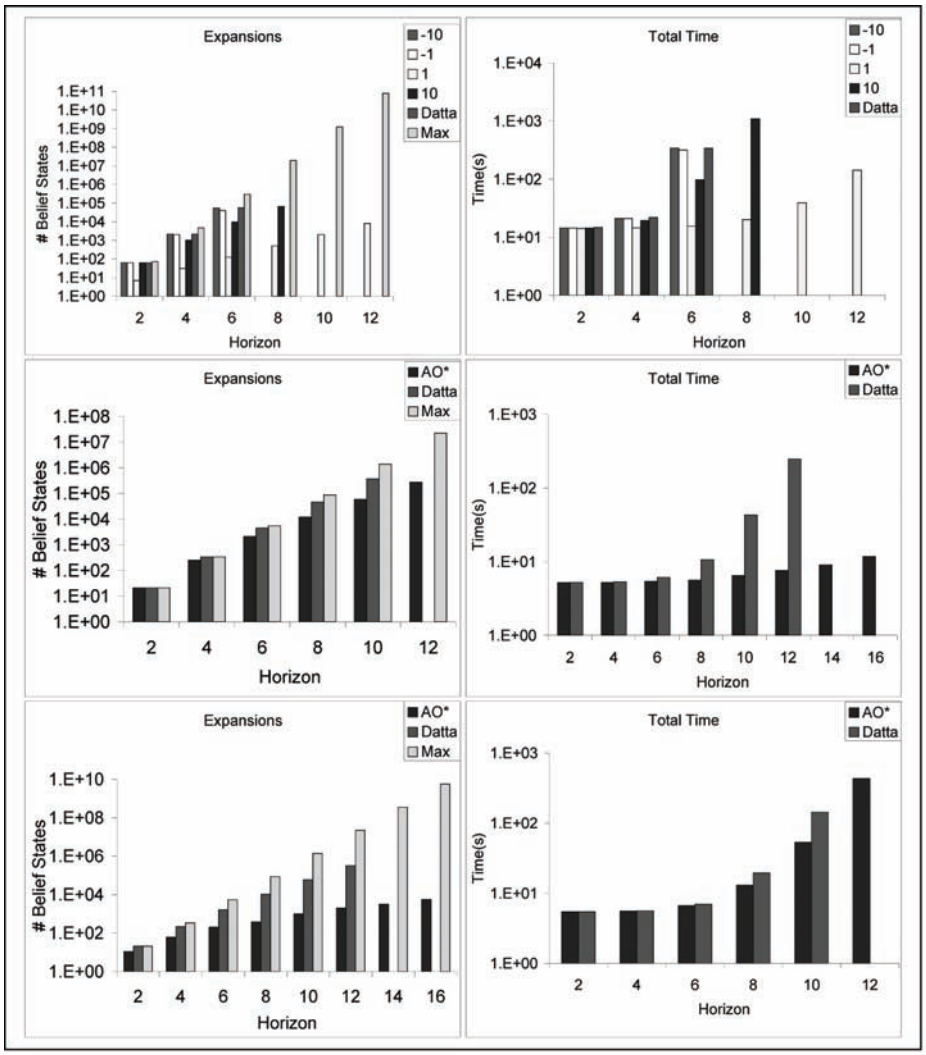
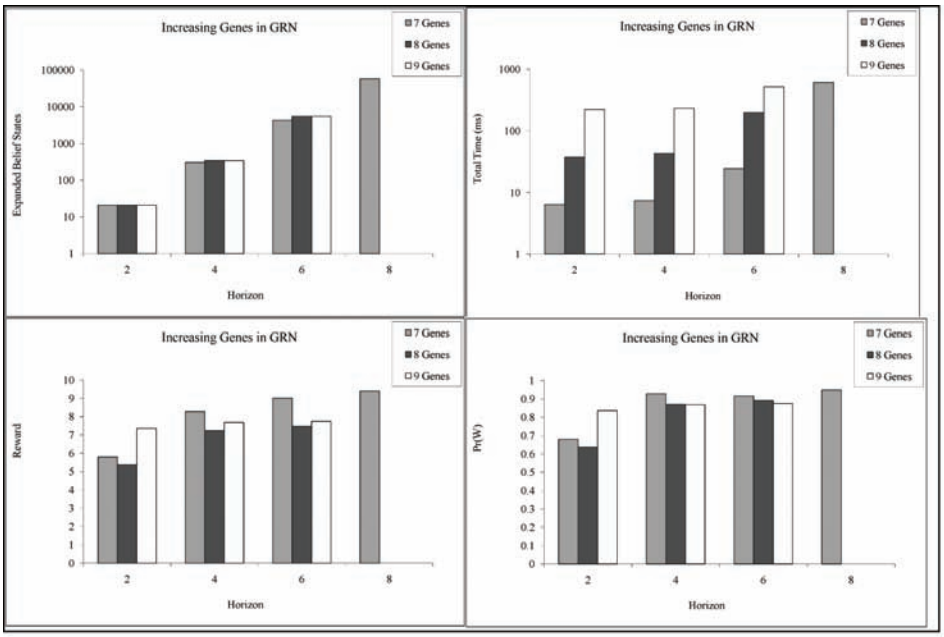


Figure 11. Increasing genes, expanded belief states (top left), total time (top right), plan reward (bottom left), and goal probability (bottom right).



and time. While the implementation of *Datta* is limited by time, the direct implementation is limited by space, exceeding memory past horizon ten. The last row of plots in Figure 10 show results in the WNT5A GRN intervening pirin and observing WNT5A. Finding plans in this problem requires more search, but *AO** can still prune.

GRN Scalability

Having established the utility of *AO** in pruning the belief state space, the next three GRNs help explore the utility of the ADD representation by increasing the number of genes from seven to nine. Figure 11 presents results of the planner solving a single intervention problem of increasing horizon in each GRN. The results depict, as before, the number of belief states expanded by *AO**, and the total time, and now include results for the reward and probability of goal satisfaction achieved in each instance.

As expected, the number of belief states expanded is the same regardless of the number genes, with the exception of a few variations due to duplicate belief states in the seven gene GRN in horizons four and six. The total time taken increases with the number of genes (because of larger ADDs) and the horizon (because of a larger belief state space). At horizon eight, only the seven gene GRN is solvable due to the extra time taken in creating the ADD representation for eight and nine gene GRNs. This indicates, as discussed further in the future research directions, that improving upon the ADD representation is necessary for increasing the number of genes. The plot of the rewards achieved by the optimal plans in each GRN indicates that the reward increases with the horizon, meaning that the length of the intervention plan does play a role in the ability to achieve the desirable states. While the data is not necessarily exhaustive, it appears that the combined reward of the plan and goals achieved slows with the horizon

indicating stable behavior in the GRN where additional reward is accrued without performing additional intervention actions. Finally, the plot of the probability of goal satisfaction indicates a tendency toward increasing probability of goal satisfaction with the horizon. In some cases, as in the seven gene network at horizon six, the plan trades the reward associated with the probability of achieving the goal for reduced intervention cost (through non-intervention). This tradeoff between plan cost and goal satisfaction is made implicit through the combined expected reward measure optimized by the plan; as explored in the future research directions, it may be useful to decouple these measures and optimize them separately.

This section has shown that AI planning is a viable avenue of research where more scalable approaches to solving GRN intervention problems may exist. Where uninformed algorithms quickly exceed time limits as the horizon increases, the more informed *AO** is sensitive to reward functions and can scale to larger horizons. Within both artificial and existing GRNs of practical interest, *AO** performs well by pruning vertices based on upper bounds. As anticipated, the interest in increasing the horizon of the plans does have a pay off in terms of both the increased probability of reaching goal states and increased expected reward.

CONCLUSION

This chapter has presented GRN intervention planning by describing techniques to derive GRNs from microarray data using the coefficient of determination, detailing the WNT5A GRN, describing the POMDP model and it applicability to capturing intervention planning. The main contribution of the chapter was the description of two alternative algorithms for solving the intervention planning problem as a POMDP and techniques used for representing the belief states and actions of the POMDP. An empirical evaluation showed that the planner (relying on several advances in AI planning to perform efficient reasoning) helped improve the scalability of planning interventions in GRNs over previous work. There remain several interesting directions for future research based on this study, and these are detailed in the following section.

FUTURE RESEARCH DIRECTIONS

There are several directions for future research based on the work presented in this chapter. Four topics are considered below: reformulating the planning problem to separate plan cost and goal achievement as separate optimization criterion, revisiting the model of time, improving scalability, and considering alternative to ADDs for the planner representation of actions and the belief state space.

Separating Action and Goal Costs

As all contemporary works on controlling GRNs, this chapter describes finding plans that optimize a single objective: the expected sum of control action costs and end state penalty costs. This can sometimes be a poor measure of the quality of an intervention plan because the cost of interventions and the cost of failure are not always expressed in the same currency. For example, when interventions costs reflect the cost of a particular drug or therapy and the failure cost reflects the cost of human life it might not make sense to combine the two into a single objective, even with a generous scaling factor. In cases

where these costs cannot be combined, we should approach the problem of planning interventions as multi-objective optimization. Computational approaches for planning with multiple objectives exist for deterministic settings (Refanidis and Vlahavas, 2003), but practical algorithms for planning with multiple objectives in stochastic settings are relatively immature.

Model of Time

The model of time in most works on controlling GRNs assumes that the rate of control matches the rate of change in the biological process (i.e., every decision point for control is after a single step of change in the biological process). In reality, when executing an intervention plan, it is not clear how much the biological process has changed when each control action is executed. The GRN is at best an approximation of the biological process, and can miss how the regulatory levels of genes change at different rates, effecting how the biological process evolves and diverging from the GRN model. The described model of GRNs already allows for uncertainty in the state of the biological process, resulting from uncertainty from the probabilistic predictor functions (i.e., regulatory mechanisms). It should be possible to strengthen the GRN model and resulting intervention plans by incorporating durational uncertainty into the predictor functions to refine state distributions. However, the data that could be used to create these distributions is currently difficult to obtain.

Improved Scalability

This chapter has shown that using heuristic search and factored representations of GRNs can help improve scalability in intervention planning, increasing the planning horizon, the number of possible interventions, the number of observations, and the number of genes. Each of these factors plays a different role in planning interventions, and there are a number of techniques to improve scalability with respect to each. The horizon, and number of observations and actions determine the size of the belief state space graph, and the number of genes determines the size of the representation of belief states. As we saw, the heuristic search algorithm *AO** used upper bounds on the possible remaining reward that can be achieved by extending the intervention plan through different belief states. These upper bounds helped determine when it was hopeless to construct the belief state space graph through a particular belief state, allowing us save considerable effort. With a perfect upper bound, it would be possible to only expand the belief state space graph that corresponds to the intervention plan, exerting the optimal amount of search effort. There are several techniques for estimating upper bounds, developed in the field of automated planning (Pearl, 1984), that could be extended to planning interventions for GRNs to help cut down search effort. Exactly computing and representing belief states can affect scalability, and we discuss several alternatives next.

Alternative Belief State Representation

Our representation of belief states, as well as actions, uses ADDs; and our ADD operations exactly compute the transitions between belief states. There are two directions that can be pursued to improve the representation and reasoning needed to generate the belief state space graph: structure and approximation. Structure is the means by which we organize the regulatory relationships between genes and their regulatory levels. An alternative structure, called a dynamic Bayesian network (DBN), is based on

Bayesian belief networks (Pearl, 1988), where we compactly represent relationships by exploiting conditional independence between genes. The DBN representation of GRNs is similar to PBNs (Shmulevich *et al.*, 2002), but somewhat more expressive. For example, in a DBN it is possible to express correlations between predictor functions for different genes. Within DBNs, there are multiple approximate inference algorithms for computing successor belief states.

Addressing the above concerns of model expressiveness and underlying representations will ultimately lead to more useful intervention plans that can address complex biological processes.

OUTLOOK

The outlook for applying automated planning to designing interventions for GRNs is promising. The goal is to scale to represent and reason with GRNs large enough to capture biologically significant systems and this could require models with hundreds to thousands of entities. Automated planning has been applied to systems of similar scale, but under the assumption of deterministic change and fully observable states. Techniques for planning with stochastic change and partially observable states are advancing to operate on a similar scale. One area of intense current focus is finding aspects of models that can be addressed by algorithms for deterministic systems and those addressed best by algorithms for stochastic systems. By decomposing models in this fashion, it is possible to use the most appropriate type of algorithms and improve scalability. Moreover, advances in algorithms developed for simulating GRNs can be integrated quite readily with planning algorithms, further improving performance.

REFERENCES

Bittner, M. (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, *406*(6795), 536–540. doi:10.1038/35020115

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research, 11*, 1–94.

Bryant, R. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, *C-35*(8), 677–691. doi:10.1109/TC.1986.1676819

Datta, A., Choudhary, A., Bittner, M., & Dougherty, E. (2004). External control in Markovian genetic regulatory networks: The imperfect information case. *Bioinformatics (Oxford, England), 20*(6), 924–930. doi:10.1093/bioinformatics/bth008

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *91*, 67–103. doi:10.1089/10665270252833208

Dougherty, E. R., Kim, S., & Chen, Y. (2000). Coefficient of determination in nonlinear signal processing. *Signal Processing*, *80*, 2219–2235. doi:10.1016/S0165-1684(00)00079-7

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*(5584), 1183–1186. doi:10.1126/science.1070919

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology, 7*(3-4), 601–620. doi:10.1089/106652700750050961

Goutsias, J., & Kim, S. (2004). A nonlinear discrete dynamical model for transcriptional regulation: Construction and properties. *Biophysical Journal*, *864*, 1922–1945. doi:10.1016/S0006-3495(04)74257-5

Goutsias, J., & Kim, S. (2006). Stochastic transcriptional regulatory systems with time delays: A mean-field approximation. *Journal of Computational Biology*, *135*, 1049–1076. doi:10.1089/cmb.2006.13.1049

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing*, 422-433.

Henkel, Z. (2007). *Investigations of the finite state linear model of gene regulatory network modeling*. MS thesis, Arizona State University, Tempe, Arizona.

Kauffman, S. (1969). Metabolic stability and epigenesist in randomly constructed genetic networks. *Journal of Theoretical Biology*, *22*, 437–467. doi:10.1016/0022-5193(69)90015-0

Kauffman, S. (1993). *The origins of order: Self-organization and selection in evolution.* New York: Oxford University Press.

Khan, S., Decker, K., Gillis, W., & Schmidt, C. (2003). A multiagent system-driven AI planning approach to biological pathway discovery. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling*.

Kim, S., Dougherty, E. R., Chen, Y., Sivakumar, K., Meltzer, P., Trent, J. M., & Bittner, M. L. (2000). Multivariate measurement of gene expression relationships. *Genomics*, *67*, 201–209. doi:10.1006/geno.2000.6241

Kim, S., Li, H., Dougherty, E., Cao, N., Chen, Y., Bittner, M., & Suh, E. (2002). Can Markov chain models mimic biological regulation? *Journal of Biological System*, *10*(4), 337–357. doi:10.1142/S0218339002000676

Lähdesmäki, H. (2003). On learning gene regulatory networks under the Boolean network model. *Machine Learning*, *52*, 147–167. doi:10.1023/A:1023905711304

Meinel, C., & Theobald, T. (1998). *Algorithms and data structures in VLSI design: OBDD–foundations and applications*. Berlin, Heidelberg, New York: Springer-Verlag.

Murphy, K., & Mian, S. (1999). *Modeling gene expression data using dynamic Bayesian networks*. (Tech. Rep.). University of California, Berkeley.

Nilsson, N. (1980). *Principles of artificial intelligence*. Morgan Kaufmann.

Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving.* Reading, MA: Addison-Wesley.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan-Kaufmann.

Refanidis, I., & Vlahavas, I. P. (2003). Multiobjective heuristic state-space planning. *Artificial Intelligence*, *145*(1-2), 1–32. doi:10.1016/S0004-3702(02)00371-5

Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics (Oxford, England)*, *18*(2), 261–274. doi:10.1093/bioinformatics/18.2.261

Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002). From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, *90*(11), 1778–1792. doi:10.1109/JPROC.2002.804686

Sosman, J. A., Weeraratna, A. T., & Sondak, V. K. (2004). When will melanoma vaccines be proven effective? *Journal of Clinical Oncology*, *22*(3), 387–389. doi:10.1200/JCO.2004.11.950

Tran, N., Baral, C., & Shankland, C. (2005). Issues in reasoning about interaction networks in cells: Necessity of event ordering knowledge. In *Proceedings of Twentieth National Conference on Artificial Intelligence*, Pittsburgh, PA (pp. 676-682).

Weeraratna, A. T., Jiang, Y., Hostetter, G., Rosenblatt, K., Duray, P., Bittner, M., & Trent, J. M. (2002). Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma. *Cancer Cell*, *1*(3), 279–288. doi:10.1016/S1535-6108(02)00045-4

ADDITIONAL READING

Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated planning: Theory and practice.* Morgan Kaufmann.

KEY TERMS AND DEFINITIONS

Dynamic Programming: A problem solving technique that extends optimal sub-solutions to optimal complete solutions.

Gene Regulatory Network: A systems biology model of intracellular components and their interactions.

Intervention: An externally controllable action that has a direct or indirect impact on the behavior of a biological system.

Melanoma: A malignant skin tumor.

Partially Observable Markov Decision Process: A formal model of extended decision making with stochastic actions and noisy observations of a partially observable environment.

Planning: Synthesizing a sequence of actions to achieve a goal.

Search: The process of evaluating alternative solutions by comparing partial or complete solutions.

Chapter 24
Mathematical Modeling of the λ Switch:
A Fuzzy Logic Approach

Dmitriy Laschov
Tel Aviv University, Israel

Michael Margaliot
Tel Aviv University, Israel

ABSTRACT

Gene regulation plays a central role in the development and functioning of living organisms. Developing a deeper qualitative and quantitative understanding of gene regulation is an important scientific challenge. The λ switch is commonly used as a paradigm of gene regulation. Verbal descriptions of the structure and functioning of the λ switch have appeared in biological textbooks. We apply fuzzy modeling to transform one such verbal description into a well-defined mathematical model. The resulting model is a piecewise-quadratic, second-order differential equation. It demonstrates functional fidelity with known results while being simple enough to allow a rather detailed analysis. Properties such as the number, location, and domain of attraction of equilibrium points can be studied analytically. Furthermore, the model provides a rigorous explanation for the so-called stability puzzle of the λ switch.

1. INTRODUCTION

Gene regulation plays a fundamental role in the development and evolution of organisms. Understanding gene regulation within living cells is a major scientific challenge in the post-genome era. Indeed, the analysis of gene regulating networks may have important implications in many fields of science, including biology and gene therapy. It may also lead to methods of synthesizing artificial networks with applications in biotechnology and biocomputing (Gardner, Cantor, & Collins, 2000).

The *λswitch* (Ptashne, 2004) is a relatively simple gene regulating network that controls two alternative patterns of gene expression in the bacterial virus λ. This epigenetic switch ensures an efficient change from one pattern to the other in response to suitable environmental cues. Bistable switches are

DOI: 10.4018/978-1-60566-685-3.ch024

common motifs in gene regulation networks, and the λ switch provides a convenient test case, as the virus is one of nature's simplest organisms. In a recent survey paper, (Zhu et al., 2007) point out that the λ switch "has indeed established itself as one of the fundamental elements in biological processes and as a paradigm for both experimental and theoretical studies in biology."

Developing suitable mathematical models for gene regulating networks is a non-trivial task. Several researchers have tried to gain a deeper understanding of the λ switch by deriving mathematical models for its dynamic behavior (see the review in Section 2.4 below). Most of the models are quite complex and, consequently, can be studied primarily using simulations and numerical analysis.

In this chapter, we apply *fuzzy modeling* (FM) to derive a new mathematical model for the λ switch. FM plays an important role in the fields of artificial intelligence and computational intelligence (Zadeh, 1994; Klir & Yuan, 1995). It is routinely used to transform the knowledge of a human expert, stated in *natural language*, into an *artificial expert system* (AES) that imitates the human expert's functioning (Siler & Buckley, 2004; Kandel, 1992). Indeed, the real power of fuzzy logic lies in its ability to handle and manipulate linguistic information based on perceptions (Dubois, Nguyen, Prade, & Sugeno, 1998; Margaliot & Langholz, 1999, 2000; Zadeh, 1996; Novak, 2005). FM provides a simple yet highly efficient approach for transforming *verbal* descriptions into well-defined mathematical models or algorithms.

Recently, FM has been used to derive mathematical models for *biological phenomena*. Biologists often provide verbal descriptions and explanations of the phenomena they study. FM provides a convenient tool for transforming these verbal descriptions into well-defined mathematical models. Note that this application of FM is somewhat different than the typical approach applied in the construction of AESs. The motivation is not to replace the human expert with an automatic algorithm, but rather to assist a human expert in transforming his/her knowledge concerning a biological phenomenon, stated in words, into a well-defined mathematical model. The usefulness of this approach was demonstrated by developing mathematical models for animal behavior (Tron & Margaliot, 2004, 2005; Bajec, Zimic, & Mraz, 2005; Rashkovsky & Margaliot, 2007; Rozin & Margaliot, 2007; Margaliot, 2007).

Fuzzy modeling of biological systems offers several advantages (Margaliot, 2008). The resulting model represents the real system in a form that corresponds closely to the way humans perceive it. Thus, the model is understandable, even by non-professionals, and each parameter has a readily perceivable meaning. The model can be easily altered to incorporate new phenomena, and if its behavior is different than expected, it is usually possible to determine which rule/term should be modified and how.

In this chapter, we apply FM to systematically transform (part of) the *verbal* description given in (Santillan & Mackey, 2004) into a mathematical model of the λ switch. The state-variables are the amounts of two regulatory proteins (*CI* and *Cro*), and the resulting model is a piecewise-quadratic second-order differential equation.

Simulations indicate that the model demonstrates adequate functional fidelity to the biological behavior. Furthermore, the piecewise-quadratic nature of the model makes it amenable to rigorous analysis. Various properties that were previously shown in simulations can now be studied analytically. These include the location and stability of the equilibrium points, and the analysis of bifurcations that may explain the *stability puzzle* in the λ switch (Santillan & Mackey, 2004).

The remainder of this chapter is organized as follows. Section 2 briefly reviews the genetic switch. Section 3 applies FM to derive a mathematical model for the λ switch. Simulations and a rigorous analysis of the mathematical model are presented in Sections 4 and 5. The final section concludes.

2. GENE REGULATION AND THE λ SWITCH

2.1. Gene Regulation

All the cells of an individual organism contain the same DNA, that is, the same genetic information. Yet, during the development of the organism from a fertilized egg, very different types of cells appear. The reason for this variety is that different genes are expressed or "turned on" in different cells. The information encoded in these genes is decoded into proteins. These proteins determine the structure and properties of the cell. Ptashne (2004) states: "At various stages, depending in part on environmental signals, cells choose to use one or another set of genes, and thereby to proceed along one or another developmental pathway."

The controlled on/off switching of sets of genes is called *gene regulation*. Analyzing the gene regulation process in high level organisms is very difficult. This is due to the large number of genes in the *DNA* and the intricate interactions between the genes. For example, the human genome contains 20,000–25,000 protein-coding genes; the *Drosophila melanogaster*, commonly known as the *fruit fly*, has approximately 14,000 protein-coding genes. It is thus natural that scientists turned their attention to gene regulation in simpler organisms. In particular, the λ phage virus, which has about 50 genes, became a prototype for studying gene regulation (Ptashne, 2004). The λ phage has been studied intensely over the last 50 years and almost all its components are now known in great detail. It is believed that developing a better understanding of the gene regulation process in the λ phage may shed light on developmental and epigenetic processes in higher organisms (Ptashne, 2004).

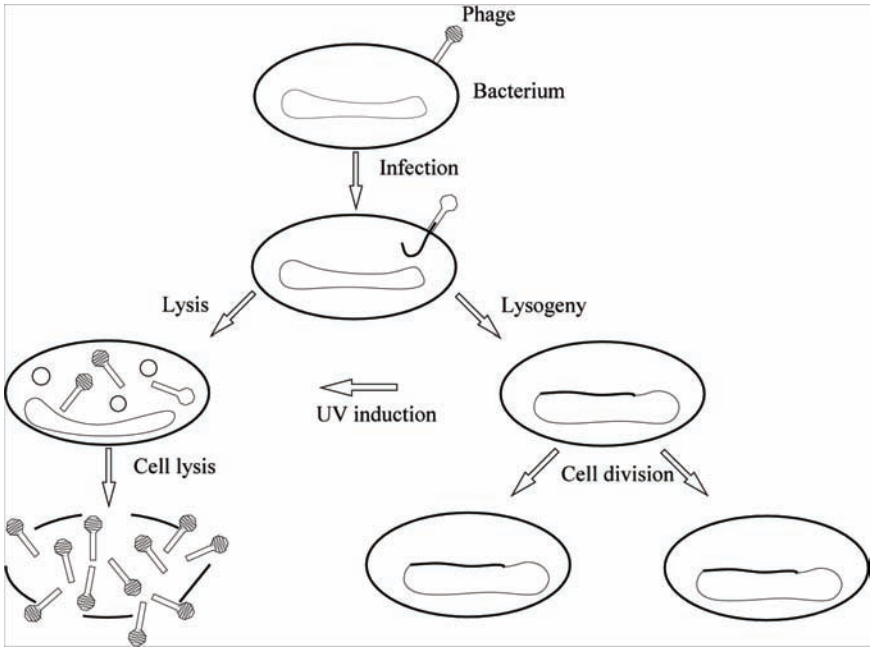
2.2. λ Phage Life Cycle

The λ phage is a virus that grows on a bacterium. The phage has a single DNA molecule. Upon infection of the bacteria, the phage injects its chromosome into the bacteria cell. The virus can then follow one of two different pathways: *lysogeny* or *lysis*.[1] In the lysogenic state, the phage integrates its genome into the bacteria's DNA and replicates as a part of the host bacterium. In the lytic state, the phage's DNA is extensively replicated, new phages are formed within the bacterium, and after about 45 minutes the bacterium lyses and releases about 100 new phages (see Figure 1).

The two possible pathways are the result of expressing different sets of genes. The phage may switch from the lysogenic state to the lytic state. This is a kind of *SOS* response initiated when the host cell experiences DNA damage. This happens, for example, if the bacteria is exposed to ultraviolet (UV) light.

The molecular mechanism responsible for the lysogeny/lysis decision is known as the *λ switch*. The λ switch has two important and striking properties. First, it is exceptionally stable. Once the lysogenic state is established, it remains stable for very long periods of time. In fact, the lysogenic state is more stable than the genome itself. The rate of mutations of the phage genome is between 10^{-6}–10^{-7} per generation, whereas the loss rate of lysogeny is less than 10^{-7} per cell and generation (Little, Shepley, & Wert, 1999; Rozanov, D'Ari, & Sineoky, 1998). Thus, the switch is robust in the sense that the probability for a random transition from one state to the other is extremely small. The second property of the switch is that it is highly efficient. In response to an appropriate signal, the phage switches to the lytic state very quickly. Thus, the switch demonstrates both stability and high switching efficiency. The coexistence of these two properties is known as the *stability puzzle*.

Figure 1. Two developmental pathways; lysogenic and lytic. UV radiation can induce lytic growth



Both these properties have a clear biological motivation. The switch should be activated only when the bacteria no longer provides a suitable host. In this case, the switch must be operated as quickly as possible, to ensure that the death of the host should not imply the death of the virus. The next section provides a simplified description of the mechanisms underlying the λ switch based on the excellent presentation in (Ptashne, 2004).

2.3. Structure and Dynamics of the λ Switch

A gene is expressed (or "on") if it is being copied by an enzyme known as *RNA polymerase*. The transcription process creates a *messenger RNA* (mRNA) which is an RNA copy of the gene. The *mRNA* molecule encodes the design of a protein.

The first step in the transcription process is the binding of *RNApolymerase* at a specific part of the gene known as the *promoter.* The *DNA* helix unwinds to produce a small open complex. One strand of DNA, the template strand, is used as a template for mRNA synthesis. As transcription proceeds, RNA polymerase traverses the template strand and uses base pairing complementarity with the DNA template to create an mRNA molecule.

The rate of transcription initiation can be related to the protein synthesis rate (Shea & Ackers, 1985).

Control of transcription initiation is the most important mechanism for determining what genes are expressed and, consequently, which proteins are produced. Regulatory proteins can either increase or decrease the probability of binding to the promoter, and thus regulate the binding process.

The total transcription rate over a period of time depends on several parameters including the concentration of transcription factors in the vicinity of the binding site, and the binding probability (sometimes referred to as *affinity*).

2.3.1. Structure of the λ Switch

The different pathways are determined by two genes: *cI* and *cro*. When *cI* is on (off) and *cro* is off (on), the phage is in the lysogenic (lytic) state. The genetic mechanisms of the λ switch concentrate along a short segment of the phage DNA which is known as the *right operator* (O_R). The operator is composed of three adjacent sites: O_R1, O_R2, and O_R3. It contains two promoters: P_R, which overlaps O_R2 and O_R1, and P_{RM}, which overlaps O_R2 and O_R3.

The transcription of the *cI* gene begins with the binding of *RNApolymerase* at P_{RM}. The resulting product is a protein *CI* that exists in two forms: monomer and dimer (denoted CI_2). In the lysogenic state, about 95% of the molecules are in the dimer form. The transcription of the *cro* gene begins with the binding of *RNApolymerase* at P_R. The product of the *cro* gene is a protein Cro_2 that exists in dimer form only.

2.3.2. Switch Dynamics

Both the CI2 and Cro2 proteins can bind to the O_R1, O_R2, and O_R3 sites and thus regulate the activity of the promoters P_R and P_{RM}. CI_2 has high affinity to O_R1, O_R2, whereas Cro_2 has high affinity to O_R3. Note that this creates a feedback loop: gene transcription yields proteins that bind to the sites regulating the transcription process.

For moderate *CI* concentration values, CI_2 binds to O_R1 and O_R2 (see Figure 2). This blocks binding of *RNApolymerase* to P_R and assists *RNA* binding to P_{RM}. Thus, the gene *cI* (*cro*) is turned on (off), yielding an increase (reduction) in the production of *CI* (Cro_2).

If *CI* concentration reaches very high values, then CI_2 binds to O_R3 as well. This blocks *RNApolymerase* binding at P_{RM} and thus reduces the production rate of *CI*. The reduction in concentration im-

Figure 2. O_R control region in the lysogenic state

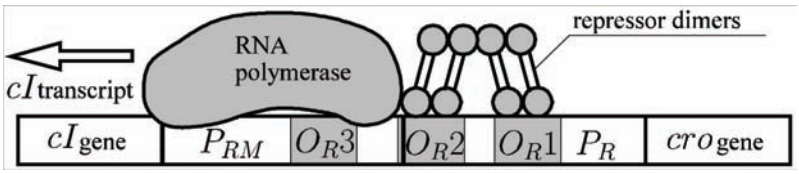
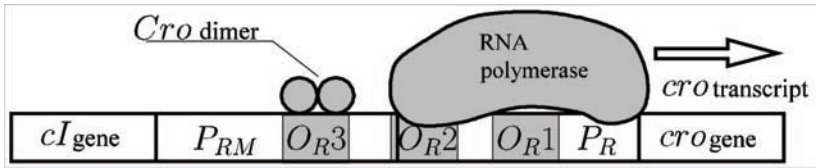


Figure 3. O_R control region in the lytic state



plies that binding at O_R3 becomes less probable (recall that CI_2 has high affinity with O_R1 and O_R2). The net result is that *CI* concentration is regulated around some high level and the phage remains in the lysogenic state.

If Cro_2 concentration is high, then it binds to O_R3 and blocks binding of *RNApolymerase* to the P_{RM} promoter (see Figure 3). This yields a reduction in the production of *CI*. If Cro_2 concentration reaches very high values, then Cro_2 binds also to O_R2 and O_R3. This blocks *RNApolymerase* binding at P_R and thus reduces the production rate of *Cro*.

2.3.3. Triggering the switch

If *CI* concentration becomes sufficiently small (for instance as a result of radiation by UV light), then *CI* binding at O_R1 and O_R2 does not take place. This leads to two results. The first is a reduction in *CI* synthesis rate (since binding of CI_2 at O_R2 helps RNA *polymerase* bind to the P_{RM} promoter). The second result is that the RNA *polymerase* can bind to P_R and thus initiate transcription of the *cro* gene (see Figure 3). This leads to an increase in Cro2 concentration. At high enough Cro2 concentrations, Cro_2 binds to O_R3 and thus further represses *CI* production and lytic growth ensues.

2.4. Mathematical Models of the λ Switch

The first quantitative model for gene regulation in the λ phage appeared in (Ackers, Johnson, & Shea, 1982). This is a statistical thermodynamic model that describes the roles of the *CI* protein and the right operator in maintaining the lysogenic state.

Shea & Ackers (1985) extended this model in order to include the effects of the other regulatory proteins, the behavior during lysogenic growth, and the induction of lysis. Under certain assumptions, they assert that a suitable model is the second-order differential

$$\begin{aligned} \dot{r}(t) &= A_r(P_1(t)k_1 + P_2(t)k_2) - d_r r(t), \\ \dot{c}(t) &= A_c P_3(t)k_3 - d_c c(t), \end{aligned} \tag{1}$$

Where r (c) is the total amount of *CI* (*Cro*) molecules in the host cell; P_1 (P_2) is the probability of *RNA* binding to P_{RM} when CI_2 is present (not present) at the O_R2 site; k_1, (k_2) is the maximal stimulated (basal) transcription rate; P_3 is the probability of *RNA* binding to the P_R promoter, and k_3 is the maximal rate of *cro* transcription; A_r (A_c) is the number of *CI* (*Cro*) molecules made per transcript; and d_r (d_c) is the degradation rate of *CI* (*Cro*). The term $P_1k_1 + P_2k_2$ thus represents the rate at which *RNA* polymerase molecules start transcription of *cI* (i.e. the rate of *RNA* isomerization from closed to open complex). Note that P_1k_1 represents the *stimulated* transcription rate due to binding of CI_2 at O_R2, and P_2k_2 represents some *basal* transcription rate. Similarly, P_3k_3 is the *cro* transcription rate.

Several studies (Shea & Ackers, 1985; Aurell & Sneppen, 2002; Santillan & Mackey, 2004) used a statistical thermodynamic approach to estimate the binding probabilities P_1, P_2, and P_3. This approach, although quite successful, has several disadvantages including: high complexity, the need for elaborate experiments, and the difficulty in specifying all the needed micro-parameters. More generally, it is not clear when equilibrium thermodynamic considerations may be used to infer high-level biological properties.

Reinitz & Vaisnys (1990) have found an inconsistency between the theoretical and experimental results of this model. McAdams & Shapiro (1995) developed a very elaborate circuit simulation model for the lysis/lysogeny decision process. Arkin, Ross, & McAdams (1998) presented a stochastic kinetic simulation of λ phage development in the very early stage, that is, right after the virus infects the bacteria. They analyzed fluctuations in gene expression rates and other molecular-level fluctuations, and their effect on the lysis/lysogeny pathway selection. Aurell & Sneppen (2002) modeled the transition between epigenetic states as a first exit problem in a dynamic system with noise, with an emphasis on stability and robustness analysis. They have found that the theoretical results do not agree with the experimental data, and concluded that the current view of the λ phage is incomplete. Santillan & Mackey (2004) extended the model developed in (Shea & Ackers, 1985) to account for some recently discovered experimental data. They also suggested an interesting explanation for the so called *stability puzzle* of the switch in terms of bifurcations induced by changes in the *CI* degradation rate. Bakk, Metzler, & Sneppen (2004) have studied the sensitivity of the right operator using new experimental data.

Several researchers have modeled gene regulation networks using piecewise linear differential equations; see, e.g. (de Jong, 2002; de Jong et al., 2004) and the references therein.

Torres and Nieto (2006) review applications of fuzzy logic in medicine and bioinformatics including problems related to gene expression. Zhu et al. (2007) provide a review of the recent theoretical and experimental results on gene regulation networks.

All existing models of the switch are quite complex and, consequently, were studied primarily using simulations. In this paper, we apply FM to develop a new mathematical model for the λ switch. The behavior of this model is congruent with known experimental results. Furthermore, the new model is simpler than previous models and is amenable to rigorous analysis.

3. FUZZY MODELLING

In this section, we use the FM approach to derive a mathematical model for the λ switch based on the verbal description of some biological observations.

A detailed presentation of FM can be found in many papers and books; see, e.g., (Klir & Yuan, 1995; Sousa & Kaymak, 2002). For the sake of completeness only, we begin by presenting the rudiments of FM using a very simple example. Readers familiar with FM may skip to Section 3.2.

3.1. Fuzzy Modeling: A Simple Example

Consider the scalar control system:

$$\dot{x}(t) = u(t),$$

where $x(t) \in R$ is the state of the system, and $u(t) \in R$ is the control. Suppose that our goal is to design a control guaranteeing that $\lim_{t\to\infty} x(t) = 0$ for any initial condition $x(0)$ It is clear that in order to achieve this, the control must be negative (positive) when $x(t)$ is positive (negative). This suggests the following two rules:

Rule 1: *if (x is positive) then u = −c,*

Rule 2: *if (x is negative) then u = c,*

Where *c* is a positive constant.

FM provides an efficient mechanism for transforming such rules into a well-defined mathematical mapping: $u = u(x)$ The first step is to define the terms in the If part of the rules. To do this, we use two functions: $\mu_{positive}(x)$ and $\mu_{negative}(x)$. Roughly speaking, for a given x, $\mu_{positive}(x)$ measures how true the proposition (*xis positive*) is. For example, we may take:

$$\mu_{positive}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x \leq 0. \end{cases}$$

However, using such a binary, 0/1, function will lead to a control that changes abruptly as x changes sign. It may thus be better to use a smoother function, say:

$$\mu_{positive}(x) = (1 + \exp(-x))^{-1}.$$

Note that now $\mu_{positive}(x)$ is a continuous function taking values in the entire interval [0,1] and satisfying: $\lim_{x\to-\infty} \mu_{positive}(x) = 0$, $\lim_{x\to+\infty} \mu_{positive}(x) = 1$. We may view $\mu_{positive}(x)$ as the *degree of membership* of x in the set of *positive numbers*. A smoother membership function seems more appropriate for sets that are defined using verbal terms. For example, consider the membership in the set of *tall people*. A small change in a person's height should not lead to an abrupt change in the degree of membership in this set.

Similarly, we may define $\mu_{negative}(x) = 1 - (1 + \exp(-x))^{-1}$. Note that this implies that:

$$\mu_{positive}(x) + \mu_{negative}(x) = 1, \text{ for all } x \in R,$$

i.e. the *total* degree of membership in the two sets is always 1.

Once the membership functions are specified, we can define the *degree of firing* (DOF) of each rule, for a given x, as $DOF_1(x) = \mu_{positive}(x)$ and $DOF_2(x) = \mu_{negative}(x)$. The output of the first (second) rule in our *fuzzy rule-base* is then defined by $-cDOF_1(x)$ ($DOF_2(x)$). In other words, the output is obtained by multiplying the DOF with the value in the Then-part of the rule. Finally, the output of the entire fuzzy rule-base is given by suitably combining the outputs of the two rules. This can be done in many ways. One standard choice is to use the so-called *center of gravity inferencing method* yielding:

$$u(x) = \frac{-cDOF_1(x) + cDOF_2(x)}{DOF_1(x) + DOF_2(x)} \cdot$$

The numerator is the sum of the rules' outputs, and the denominator plays the role of a scaling factor. Note that we may also express this as:

$$u(x) = -c\frac{DOF_1(x)}{DOF_1(x)+DOF_2(x)} + c\frac{DOF_2(x)}{DOF_1(x)+DOF_2(x)},$$

which implies that the output is always a convex combination of the rules' outputs.

Substituting the membership functions yields the controller:

$$\begin{aligned} u(x) &= -c(1+\exp(-x))^{-1} + c(1-(1+\exp(-x))^{-1}) \\ &= -c\tanh(x/2). \end{aligned}$$

Note that this can be viewed as a smooth version of the controller:

$$u(x) = \begin{cases} -c, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ c, & \text{if } x < 0. \end{cases}$$

Summarizing, FM allows us to transform verbal information, stated in the form of If-Then rules, into a well-defined mathematical function. Note that the fuzziness here stems from the inherent vagueness of verbal terms. This vagueness naturally implies that any modeling process based on verbal information would include many degrees of freedom (Margaliot, 2008). Yet, it is important to note that the final result of the FM process is a completely well-defined mathematical formulation.

3.2. Fuzzy Modeling of the λ Switch

Following (Shea & Ackers, 1985), we consider a mathematical model of the form:

$$\begin{aligned} \dot{r}(t) &= A_r a_r(t) - d_r r(t), \\ \dot{c}(t) &= A_c a_c(t) - d_c c(t), \end{aligned} \tag{2}$$

Where r (c) is the total amount of *CI* (*Cro*) molecules in the host cell; a_r, a_c are the transcription rates; A_r (A_c) is the number of *CI* (*Cro*) molecules made per transcript; and d_r (d_c) is the degradation rate of *CI* (*Cro*). Note that this definition implies that all the parameters and variables are positive. Note also that (2) is exactly the model (1) with:

$$\begin{aligned} a_r(t) &= a_s(t) + a_b(t) = k_1 P_1(t) + k_2 P_2(t) \\ a_c(t) &= k_3 P_3(t). \end{aligned}$$

Specifying the time-varying transcription rates $a_r(t)$ and $a_c(t)$ is the most difficult part in the modeling process. We use FM in order to determine these rates, and thus complete the mathematical model (2).

The application of FM in biological systems is based on a verbal description of the biological phenomena. This is transformed into a set of fuzzy If-Then rules. Suitable membership functions are used to define the verbal terms in these rules. Inferencing produces a well-defined mathematical model (Tron & Margaliot, 2004).

3.3. Verbal Description of the λ Switch

The λ phage has been intensely studied over the last 50 years and many verbal descriptions of the λ switch exist. The most suitable for our purposes is the following description from (Santillan & Mackey, 2004):

"…dimmers CI_2 repress the production of Cro and enhance the production of CI Nevertheless, if the concentration of CI_2 reaches very high values, the probability for CI_2 to bind O_R3 will be increased, which has the effect of repressing RNA polymerase binding to P_{RM}. Thus CI_2 regulates its own concentration by enhancing CI production if its concentration is not too high, and otherwise repressing transcription of gene cI If the CI_2 concentration decreases, for instance by the cleavage of CI by RecA proteins (activated by UV light), the probability for O_R1 and O_R2 to be free from CI_2 is increased. This, on its own, creates the possibility that a polymerase will bind P_R and start transcription of gene cro and, in the long run, leads to an increasing Cro_2 concentration. At a high enough Cro_2 concentration, a Cro_2 can bind O_R3 and repress CI production, establishing the lytic state. In this state, gene cro is on while gene cI is off. When the concentration of Cro_2 is too high, a Cro_2 can bind to O_R2 and even to O_R1, repressing the production of Cro."

The first stage in the FM approach is transforming the given verbal description into appropriate fuzzy rules.

3.4. Fuzzy Rules

Our first set of rules describes the *cI* transcription rate a_r:

Rule 1: *if (r is $medium_r$) and (c is low_c) then $a_r = a_1$.*

Rule 2: *if (r is low_r) and (c is low_c) then $a_r = a_2$.*

Rule 3: *if (r is $high_r$) or (c is $high_c$) then $a_r = 0$.*

Here $a_1 > a_2 > 0$. The subscripts r, c in the verbal terms, such as in low_r and low_c, are used to distinguish between terms that will later be modeled using different functions. Rule 1 corresponds to the case where *CI* concentration is medium and *Cro* concentration is low, so *CI* binds to both O_R1 and O_R2, yielding the maximal *cI* transcription rate a_1. Rule 2 corresponds to the case where the concentrations of both proteins are low, and the O_R sites are free, yielding the basal transcription rate a_2. The last rule describes the situation where either *CI* concentration is high (so it also binds to O_R3), or the *Cro* concentration is high and Cro_2 binds to the O_R sites. In both cases, the transcription of the *cI* gene is suppressed.

The rules describing the *cro* transcription rate are:

Rule 4: *if (r is low_r) and (c is $not\_very\_high_c$) then* $a_c = a_3$.

Rule 5: *if (r is $medium_r$) or (r is $high_r$) or (c is $very\_high_c$) then* $a_c = 0$.

Here $a_3 > 0$, Rule 5 describes the situation where the transcription of *Cro* is suppressed. This happens when either: (1) *CI* concentration is medium, so *cro* is turned off; or (2) *Cro* concentration is very high, so it binds to O_R1 or O_R2. Rule 4 describes the complementary situation yielding the maximal *cro* transcription rate a_3.

The next step is to model the verbal terms in the rules (e.g., low_c) using suitable membership functions.

3.5. Fuzzy Membership Functions

Fuzzy terms such as "$rismedium_r$" are modeled using *membership functions*, that is, a function $\mu_{medium\_r}(r)$ mapping the domain of possible r values to [0,1]. For a given r, $\mu_{medium\_r}(r)$ models the degree of membership of r in the set of "medium values". We say that the membership function is *normal* if there exists a value s such that $\mu_{medium}(s) = 1$.

We use *piecewise linear* (PWL) membership functions. These can be used to provide accurate approximation of arbitrary smooth functions and, as we will see below, lead to a mathematical model that is amenable to analysis.

For two vectors $\alpha, \beta \in R^n$, with $\alpha_1 < \alpha_3 < \ldots < \alpha_n$ and $\beta_i \in [0,1]$ for $i = 1, \ldots, n$, let $s(\cdot; \alpha, \beta) : R \to R$ denote the PWL function such that $s(\alpha_i; \alpha, \beta) = \beta_i$. In other words, s linearly interpolates between the points (α_i, β_i), $i = 1, \ldots, n$. The function $1 - s(\cdot; \alpha, \beta)$ is also PWL, and for the sake of notational convenience, we let $\bar{\beta} \in R^n$ denote the vector such that $s(\cdot; \alpha, \bar{\beta}) = 1 - s(\cdot; \alpha, \beta)$.

Our fuzzy rules include seven verbal terms: low_r, $medium_r$, $high_r$, low_c, $high_c$, $very\_high_c$, and $not\_very\_high_c$. We model the terms characterizing r using membership functions in the form:

$$\begin{aligned}
\mu_{low_r}(r) &= s(r; \alpha^1, \beta^1), \\
\mu_{high_r}(r) &= s(r; \alpha^1, \beta^2), \\
\mu_{medium_r}(r) &= 1 - (\mu_{low_r}(r) + \mu_{high_r}(r)).
\end{aligned} \tag{3}$$

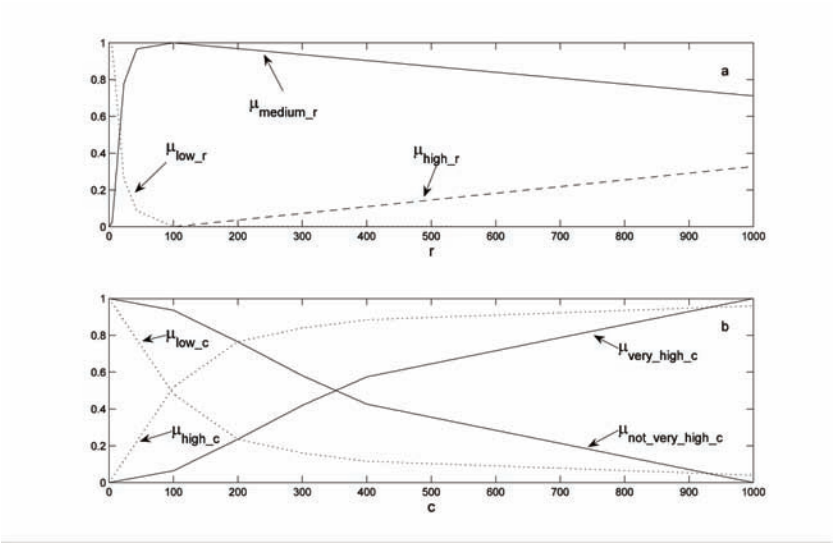
Note that this implies that $\mu_{low\_r}(r) + \mu_{medium\_r}(r) + \mu_{high\_r}(r) = 1$ for all r Since r can attain either a *low*, *medium*, or *high* value, the sum of its degrees of belonging to all three sets should be 1. Eq. (3) yields

$$\begin{aligned}
\mu_{medium_r}(r) &= 1 - (s(r; \alpha^1, \beta^1) + s(r; \alpha^1, \beta^2)) \\
&= 1 - s(r; \alpha^1, \beta^3) \\
&= s(r; \alpha^1, \bar{\beta}^3),
\end{aligned}$$

where $\beta^3 := \beta^1 + \beta^2$.

The verbal terms for c are defined using:

Figure 4. Membership functions; (a) for r, (b) for c



$$\begin{aligned}
\mu_{high_c}(c) &= s(c;\alpha^2,\beta^4),\\
\mu_{low_c}(c) &= 1-\mu_{high_c}(c) = s(c;\alpha^2,\overline{\beta^4}),\\
\mu_{very_high_c}(c) &= s(c;\alpha^2,\beta^5),\\
\mu_{not_very_high_c}(c) &= 1-\mu_{very_high_c}(c) = s(c;\alpha^2,\overline{\beta^5}).
\end{aligned} \tag{4}$$

Figure 4 depicts a schematic view of all the membership functions. The next step in the FM approach is fuzzy inferencing.

3.6. Fuzzy Inferencing

We use the *center of gravity* inferencing method, and product (sum) for the logical *and* (*or*) operator (Sousa & Kaymak, 2002). The first three fuzzy rules yield:

$$a_r(r,c) = \frac{a_1\mu_{medium\_r}(r)\mu_{low\_c}(c) + a_2\mu_{low\_r}(r)\mu_{low\_c}(c)}{D_r}, \tag{5}$$

where

$$\begin{aligned}
D_r &:= \mu_{medium_r}(r)\mu_{low_c}(c) + \mu_{low_r}(r)\mu_{low_c}(c)\\
&+\mu_{high_r}(r) + \mu_{high_c}(c).
\end{aligned}$$

Using the definition of the membership functions yields: $D_r = 1 + \mu_{high\_c}(c)\mu_{high\_r}(r)$.

Since the two proteins suppress each other, we may assume that the system is never in a state where both *CI* and *Cro* concentrations are high, so $\mu_{high\_c}(c)\,\mu_{high\_r}(r) \approx 0$. Thus, we simplify (5) to

$$a_r(r,c) = a_1\mu_{medium\_r}(r)\mu_{low\_c}(c) + a_2\mu_{low\_r}(r)\mu_{low\_c}(c). \tag{6}$$

Note that the first (second) term on the right-hand side of this equation can be interpreted as the stimulated (basal) transcription rate.

Similarly, Rules 4 and 5 yield

$$a_c(r,c) = \frac{a_3\mu_{low\_r}(r)\mu_{not\_very\_high\_c}(c)}{D_c}, \tag{7}$$

where: $D_c := \mu_{low\_r}(r)\mu_{not\_very\_high\_c}(c) + \mu_{medium\_r}(r) + \mu_{high\_r}(r) + \mu_{very\_high\_c}(c)$. Using the definition of the membership functions yields:

$$D_c = 1 + \mu_{medium\_r}(r)\mu_{very\_high\_c}(c) + \mu_{high\_r}(r)\mu_{very\_high\_c}(c).$$

Arguing as above yields $D_c \approx 1$, so we simplify (7) to:

$$a_c(r,c) = a_3\mu_{low\_r}(r)\mu_{not\_very\_high\_c}(c). \tag{8}$$

Substituting (6) and (8) in (2) yields:

$$\begin{aligned} \dot{r} &= A_r(a_1\mu_{medium\_r}(r)\mu_{low\_c}(c) + a_2\mu_{low\_r}(r)\mu_{low\_c}(c)) - d_r r, \\ \dot{c} &= A_c a_3\mu_{low\_r}(r)\mu_{not\_very\_high\_c}(c) - d_c c. \end{aligned} \tag{9}$$

Note that since the membership functions are PWL, (9) is a *piecewise-quadratic second-order differential equation.*

3.7. Parameter Estimation

To complete the model (9), we need to specify the parameters $A_r, A_c, a_1, a_2, a_3, d_r, d_c$, and the four membership functions $\mu_{low\_c}(c)$, $\mu_{not\_very\_high\_c}(c)$, $\mu_{low\_r}(r)$, $\mu_{medium\_r}(r)$. To do so, we use some of the experimental data reported in (Shea & Ackers, 1985; Bakk et al., 2004; Hawley & McClure, 1982).

In particular, (Bakk et al., 2004) provide $a_r(r, 0)$, that is, the *cI* transcription rate as a function of *CI* in the absence of *Cro*. In this case, $\mu_{small\_c}(c) = 1$, so (6) yields:

$$a_r(r,0) = a_1\mu_{medium\_r}\left(r\right) + a_2\mu_{low\_r}\left(r\right). \tag{10}$$

We can view the first (second) term on the right-hand side as the stimulated (basal) transcription rate, so a comparison with (1) yields:

$$a_1\mu_{medium\_r}(r) = k_1 P_1(r,0),$$
$$a_2\mu_{low\_r}(r) = k_2 P_2(r,0). \tag{11}$$

Thus, we set $a_1 = k_1 \max_r P_1(r,0)$, $a_2 = k_2 \max_r P_2(r,0)$, and design the membership functions such that:

$$\mu_{medium\_r}(r) = s(r;\alpha^1,\overline{\beta^3}) = \frac{P_1(r,0)}{\max_r P_1(r,0)}, \quad \mu_{low\_r}(r) = s(r;\alpha^1,\beta^1) = \frac{P_2(r,0)}{\max_r P_2(r,0)}. \tag{12}$$

This guarantees that the membership functions are normal. Summarizing, determining the membership functions is done by constructing PWL approximations of the functions $P_1(r,0) / \max_r P_1(r,0)$, and $P_2(r,0) / \max_r P_2(r,0)$.

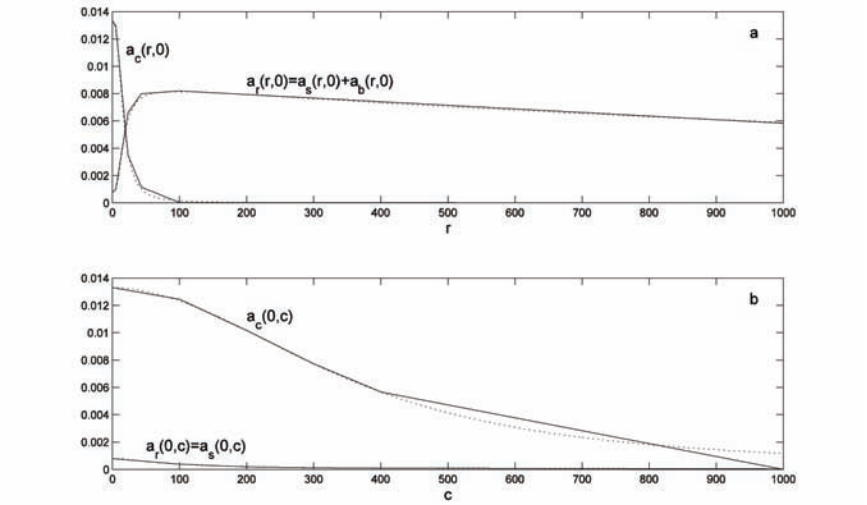
Figure 5 (*a*) depicts the transcription rates as a function of *r* in the (Bakk et al., 2004) model (dotted lines) and in our PWL approximation.

Using a similar technique for the case *r* = 0, we derived the parameters for $s(c;\alpha^2,\overline{\beta^4})$ and $s(c;\alpha^2,\overline{\beta^5})$ (see Figure 5 (*b*)). All the model parameters are summarized in the Appendix.

Substituting (3) and (4) in (6) and (8) yields the transcription rates:

$$a_r(r,c) = s(c;\alpha^2,\overline{\beta^4})(a_1 s(r;\alpha^1,\overline{\beta^3}) + a_2 s(r;\alpha^1,\beta^1)),$$
$$a_c(r,c) = a_3 s(r;\alpha^1,\beta^1) s(c;\alpha^2,\overline{\beta^5}). \tag{13}$$

Figure 5. Transcription rates $a_r(r,c)$ and $a_c(r,c)$ in our model (solid lines) and in the (Bakk et al., 2004) model (dotted lines). (a) as a function of r when c = 0; (b) as a function of c when r = 0



Since s is PWL, this implies that the r-c domain is divided into cells, denoted C^{ij}, such that for $(r, c) \in C^{ij}$:

$$\begin{aligned} a_r(r,c) &= rcp_1^{ij} + rp_2^{ij} + cp_3^{ij} + p_4^{ij}, \\ a_c(r,c) &= rcp_5^{ij} + rp_6^{ij} + cp_7^{ij} + p_8^{ij}. \end{aligned} \tag{14}$$

The number and topology of these cells, and the values of the constants p_1^{ij},& p_8^{ij}, follow immediately from the parameters of the PWL membership functions. For example, one such cell is:

$$C^{11} = \{(r,c) : \alpha_1^1 \le r \le \alpha_2^1, \alpha_1^2 \le c \le \alpha_2^2\},$$

and for $(r, c) \in C^{11}$, Eq. (13) becomes:

$$\begin{aligned} a_r(r,c) &= \frac{(\bar{\beta}_2^4 - \bar{\beta}_1^4)c + \bar{\beta}_1^4\alpha_2^2 - \bar{\beta}_2^4\alpha_1^2}{\alpha_2^2 - \alpha_1^2} \\ &\quad \times \left(a_1 \frac{(\bar{\beta}_2^3 - \bar{\beta}_1^3)r + \bar{\beta}_1^3\alpha_2^1 - \bar{\beta}_2^3\alpha_1^1}{\alpha_2^1 - \alpha_1^1} + a_2 \frac{(\beta_2^1 - \beta_1^1)r + \beta_1^1\alpha_2^1 - \beta_2^1\alpha_1^1}{\alpha_2^1 - \alpha_1^1} \right), \\ a_c(r,c) &= a_3 \frac{(\beta_2^1 - \beta_1^1)r + \beta_1^1\alpha_2^1 - \beta_2^1\alpha_1^1}{\alpha_2^1 - \alpha_1^1} \times \frac{(\bar{\beta}_2^5 - \bar{\beta}_1^5)c + \bar{\beta}_1^5\alpha_2^2 - \bar{\beta}_2^5\alpha_1^2}{\alpha_2^2 - \alpha_1^2}. \end{aligned}$$

Note that this implies that the transcription rates are piecewise-quadratic functions.

Substituting (14) in (2) yields:

$$\begin{aligned} \dot{r} &= A_r\left(rcp_1^{ij} + rp_2^{ij} + cp_3^{ij} + p_4^{ij}\right) - rd_r, \\ \dot{c} &= A_c\left(rcp_5^{ij} + rp_6^{ij} + cp_7^{ij} + p_8^{ij}\right) - cd_c, \end{aligned} \tag{15}$$

with $i, j \in \{1, \ldots, 5\}$. The p_k^{ij} values are listed in Tables 3 and 4 in the Appendix.

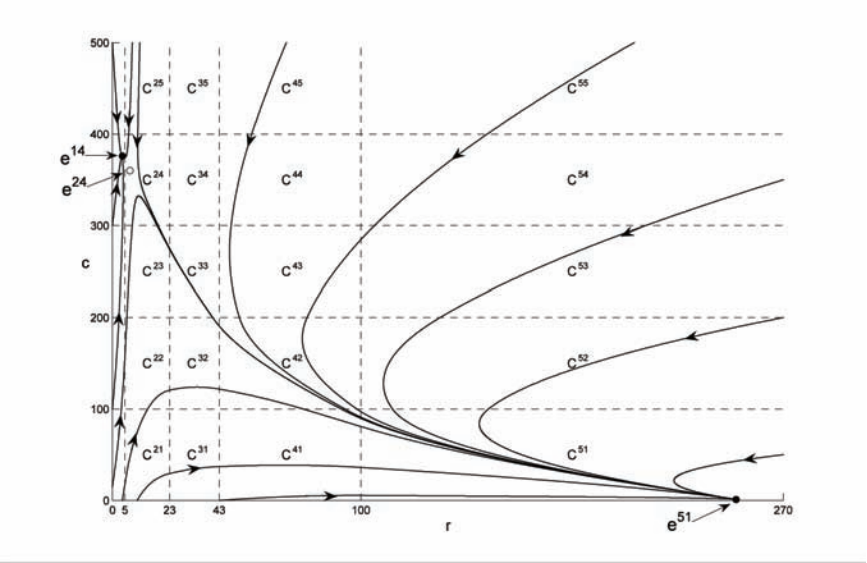
This completes the derivation of the mathematical model. Note that (15) is a piecewise-quadratic second-order differential equation. In the following sections, we study the behavior of (15) using both simulations and rigorous analysis.

4. SIMULATIONS

Figure 6 depicts phase space trajectories of (15). These were obtained by numerically solving the differential equation (15) for several initial conditions ($r(0)$, $c(0)$). Cell boundaries, denoted by dashed lines, are also shown.

It may be seen that the system admits at least two equilibrium points, one in cell C^{51} (denoted e^{51}), and one in cell C^{14} (denoted e^{14}). $e^{51} \approx (251, 1)$ corresponds to a steady-state where r is high and c is low

Figure 6. Phase space trajectories in the (r,c) plane



(i.e., the lysogenic state), and is locally asymptotically stable with a relatively large domain of attraction. $e^{14} \approx (4, 376)$ corresponds to the lytic state, and has a smaller domain of attraction.

Summarizing, the simulations indicate that under normal conditions, the switch admits two stable steady-states corresponding to the lysogenic and lytic states. It is important to note that the qualitative value $e^{51} \approx (251, 1)$ agrees with known experimental data for the total protein levels in the lysogenic state (Shea & Ackers, 1985; Aurell & Sneppen, 2002; Ptashne 2004).

In the next section, we rigorously analyze (15) in order to confirm the behavior depicted in Figure 6.

5. ANALYSIS

5.1. Equilibrium Points

For a given cell, C^{ij}, (15) admits an equilibrium point if the equations:

$$\begin{aligned} 0 &= A_r\left(rcp_1^{ij} + rp_2^{ij} + cp_3^{ij} + p_4^{ij}\right) - rd_r, \\ 0 &= A_c\left(rcp_5^{ij} + rp_6^{ij} + cp_7^{ij} + p_8^{ij}\right) - cd_c, \end{aligned} \tag{16}$$

admit a solution $e = (r,c)$, satisfying $e \in C^{ij}$. The first equation implies that $r = \dfrac{cp_3^{ij} + p_4^{ij}}{(d_r / A_r) - cp_1^{ij} - p_2^{ij}}$, and substituting this in the second equation yields: $0 = \left(cp_3^{ij} + p_4^{ij}\right)\left(cp_5^{ij} + p_6^{ij}\right) + \left(cp_7^{ij} + p_8^{ij} - (d_c / A_c)c\right)\left((d_r / A_r) - cp_1^{ij} - p_2^{ij}\right)$, which is a quadratic equation in c. These equations are thus easily solved.

For our parameter values, (15) admits three equilibrium points: $e^{14} = (4, 376) \in C^{14}$, $e^{24} = (7, 360) \in C^{24}$, and $e^{51} = (251, 1) \in C^{51}$ (the numerical values are rounded to the nearest integer). These values agree with experimental data (Ptashne, 1986, 1992). Linearization about the equilibrium points shows that e^{14} and e^{51} are locally asymptotically stable (LAS), whereas e^{24} is a saddle point.

5.2. Limit Cycles

Since the differential equation in each cell C^{ij} is quadratic, it is possible in principle that either C^{14} or C^{51} contain a limit cycle. It follows from (15) that in cell C^{ij}:

$$\frac{d\dot{r}}{dr} + \frac{d\dot{c}}{dc} = A_r(cp_1^{ij} + p_2^{ij}) - d_r + A_c(rp_5^{ij} + p_7^{ij}) - d_c,$$

so the curve $Z^{ij} := \{(r, c) \in C^{ij} : \frac{d\dot{r}}{dr} + \frac{d\dot{c}}{dc} = 0\}$ is a line. By the Bendixon-Dulac theorem (see, e.g., (Zhi-fen, Tong-ren, Wen-zao, & Zhen-xi, 1992, p. 195)), any closed trajectory must intersect the line Z^{ij}. It is easy to verify that for the parameter values in our model $Z^{ij} \not\in C^{ij}$ for both $(i, j) = (1, 4)$ and $(i, j) = (5, 1)$. Thus, we conclude the following.

Proposition 1. None of the cells C^{ij}, $i, j \in \{1, \ldots, 5\}$, contains a closed trajectory.

5.3. Domain of Attraction

Consider a differential equation:

$$\dot{x} = f(x), \tag{17}$$

where $f : R^n \to R^n$ is a smooth vector field. Let $x(t; x_0)$ denote the solution of (17) at time $t \geq 0$ for the initial condition $x(0) = x_0$.

Definition 1. If e is an asymptotically stable equilibrium point of (17), then its *domain of attraction* is the set: $D(e) := \{y \in R^n : \lim_{t\to\infty} x(t; y) = e\}$.

In other words, any solution emanating from $D(e)$ converges to e. The size and shape of the attraction domain provide important information on the behavior of the dynamic system. Our goal in this section is to obtain an estimation of the attraction domains $D(e^{51})$ and $D(e^{14})$. We denote the right-hand side of (15) by f^{ij} i.e. $f^{ij} \in R^2$ is the vector field in cell C^{ij}

Definition 2. A cell $C^{ij} = \{(r, c) : \alpha_i^1 \leq r \leq \alpha_{i+1}^1, \alpha_j^2 \leq c \leq \alpha_{j+1}^2\}$ is called *absorbing* if the following four conditions hold:

$\dot{r}(\alpha_i^1, c) \geq 0$, for all $c \in [\alpha_j^2, \alpha_{j+1}^2]$

$\dot{r}(\alpha_{i+1}^1, c) \leq 0$, for all $c \in [\alpha_j^2, \alpha_{j+1}^2]$

$\dot{c}(r, \alpha_j^2) \geq 0$, for all $r \in [\alpha_i^1, \alpha_{i+1}^1]$

$\dot{c}(r, \alpha_{j+1}^2) \leq 0$, for all $r \in [\alpha_i^1, \alpha_{i+1}^1]$

In other words, on the boundary of C^{ij}, f^{ij} points inside C^{ij}

It follows from this definition that if C^{ij} is absorbing, then any solution emanating from C^{ij} remains in C^{ij} for all $t \geq 0$. By the Poincare-Bendixon theorem (Zhi-fen, Tong-ren, Wen-zao, & Zhen-xi, 1992), the ω-limit set of such a trajectory is either an equilibrium point or a limit cycle. Combining this with Proposition 1 yields the following result.

Proposition 2. Any absorbing cell C^{ij} in our model contains a LAS equilibrium point e^{ij} and C^{ij} Í $D(e^{ij})$.

Definition 3. Two cells $C^{ij}$and C^{pq} are called *contiguous* if $C^{ij} \cap C^{pq} \neq \varnothing$.

Definition 4. A cell C^{pq} is said to be a *transition to cell* C^{ij}if C^{ij} and C^{pq} are contiguous, and for any $x_0 \in C^{pq}$ there exists a time $t > 0$ such that x(t; x_0) $\in C^{ij}$.

In other words, any solution emanating from C^{pq} reaches C^{ij} at some time $t > 0$.

The next result provides a sufficient condition for a cell to be a transition cell based on the direction of the vector field along the boundaries of the cell. We use ∂C^{pq} to denote the boundary of cell C^{pq}.

Proposition 3. A cell C^{pq} is a transition to cell C^{ij} if the following conditions hold. (1) C^{pq} does not contain any equilibrium points or limit cycles; (2) C^{ij} and C^{pq} are contiguous; (3) for any $x \in C^{pq}$ Ç C^{ij}, the vector field $f^{pq}(x)$ points from C^{pq} to C^{ij}; and (4) for any $x \in \partial C^{pq} \setminus (C^{ij}$Ç $C^{pq})$, $f^{pq}(x)$ points into C^{pq} .

The proof is immediate: by condition (1), any solution satisfying $x(0) \in C^{pq}$ must leave C^{pq} at some time $t > 0$ The conditions on f^{pq} imply that it can only leave to C^{ij}

Proposition 3 can be used, for example, to prove that C^{41} is a transition to C^{51} (see Figure 6). The next result follows immediately from the above definitions.

Proposition 4. Suppose that C^{ij} is absorbing and let $e^{ij} \in C^{ij}$ be the equilibrium point which exists by Proposition 2. If C^{pq} is a transition to C^{ij}, then C^{pq} Í $D(e^{ij})$. Furthermore, any cell that is a transition to C^{pq} is also contained in $D(e^{ij})$.

This simple result provides an iterative recipe for constructing an estimate for $D(e^{ij})$.

5.3.1. Estimating $D(e^{51})$

We now use the results above in order to derive an estimation for $D(e^{51})$, that is, the attraction domain of the lysogenic state. It is straightforward to prove (using the parameters given in the Appendix), that C^{51} is absorbing. By Proposition 2, it contains an LAS equilibrium point e^{51}, and $C^{51} \in D(e^{51})$. Using Proposition 3 shows that C^{41} is a transition to C^{51}, so $C^{41} \in D(e^{51})$.

The next step is to prove that C^{42} and C^{52} also belong to $C^{41} \in D(e^{51})$. This, however, does not follow immediately from Proposition 3, since neither of these cells is a transition to C^{41} nor to C^{51} (see Figure 7). It can be shown, however, that C^{52} can be divided into two sub-cells: C_1^{52} and C_2^{52} such that C_1^{52} is a transition to C^{51} (and, therefore, $C_1^{52} \in D(e^{51})$), and any trajectory emanating from C^{42} reaches either C^{41} or C_1^{52}. This proves that $C^{42} \in D(e^{51})$ (see Figure 7). Proceeding in this fashion yields the following result.

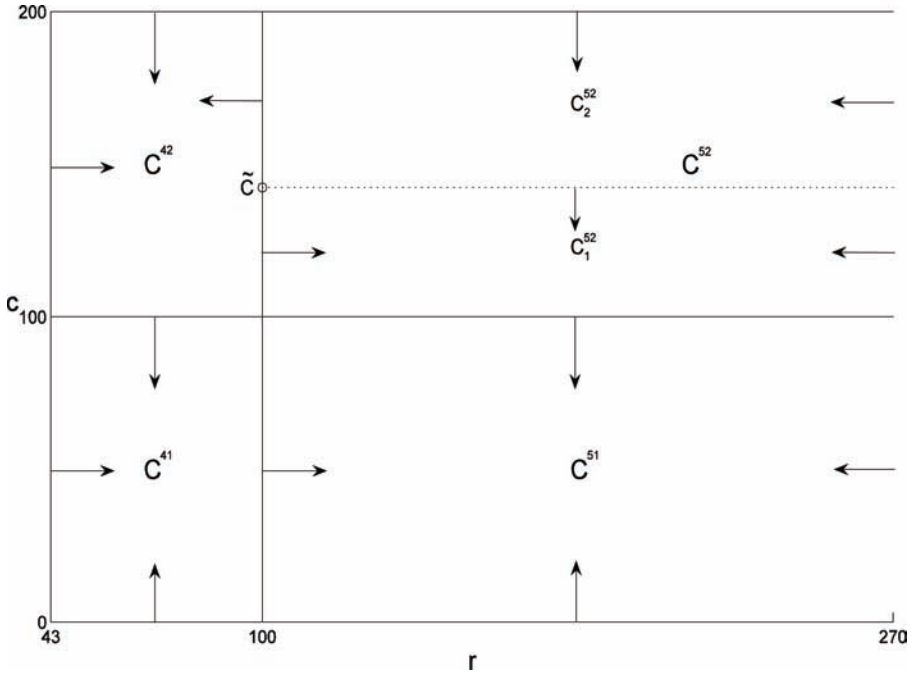
Proposition 5. $D(e^{51})$ contains the cells: C^{31}, C^{32}, C^{33}, C^{34}, C^{41}, C^{42}, C^{43}, C^{44}, C^{51}, C^{52}, C^{53}, and C^{54}.

This large attraction domain suggests that e^{51} is "very stable" in the sense that a large magnitude perturbation is needed in order to force the system away from $D(e^{51})$.

5.3.2. Estimating $D(e^{14})$

It is easy to verify that C^{14} is absorbing and, therefore, contains an equilibrium point e^{14}.

Figure 7. Cell transitions in the neighborhood of e^{51}



Proposition 3 can be used to show that C^{15} is a transition to C^{14} (see Figure 8), and Proposition 4 implies that $C^{15} \in D(e^{14})$. Thus, we proved the following.

Proposition 6. $D(e^{14})$ contains C^{14} and C^{15}.

5.4. Activating the Switch

For the parameter values described above, the system admits two LAS equilibrium points: e^{51} and e^{14} corresponding to the lysogenic and lytic states, respectively. The system also admits a saddle point, e^{24}, which is located near e^{14}. The relatively large domain of attraction $D(e^{51})$ may explain the robustness of the lysogenic state with respect to random perturbations.

However, the robustness of e^{51} also suggests that the transition from lysogeny to lysis must be relatively slow, and this does not agree with the very fast switching behavior which takes place during a SOS response. The SOS response can be triggered by exposing the host bacteria to UV radiation. Cleavage by RecA proteins then leads to a decrease in *CI* concentration.

Santillan & Mackey (2004) presented an interesting mathematical model for the λ switch. They used simulations to show that increasing the *CI* degradation rate d_r yields bifurcations in the dynamic model. For a sufficiently large value of d_r, all equilibrium points disappear, except for the one corresponding to the lytic state, which becomes globally asymptotically stable. Thus, bistability is lost and all initial conditions converge to the lytic state. We now show that our model displays a similar behavior.

5.4.1. Simulations

Let $d_n := 1/2943$ denote the nominal value of the *CI* degradation rate (see Table 2). Figure 9 depicts the phase space trajectories of our model for $d_r = 4d_n$ It may be seen that there exists a single equilibrium point, e^{14}, corresponding to the lytic state, and that e^{14} is globally asymptotically stable.

Figure 10 depicts the state space trajectories for the intermediate value $d_r = 2d_n$. It may be seen that the equilibrium point e^{51} and the saddle point move toward each other. For some value $d_r \in (2d_n, 4d_n)$, these two points disappear.[2] Bistability is lost, and the lytic state becomes globally asymptotically stable.

Thus, our model can be used to explain the stability puzzle in a similar way as does the model in (Santillan & Mackey, 2004). Note, however, that the model in (Santillan & Mackey, 2004) is a fourth-order differential equation with time-delay, and is therefore more complicated than our model. Indeed, our model seems simple enough to rigorously analyze the bifurcations. In particular, in our model the analysis of equilibrium points is reduced to studying the roots of a quadratic equation. In the next section, we analyze this issue in more detail.

5.4.2. Analysis

In this section, we analyze the effect of increasing d_r on the equilibrium point corresponding to the lytic state. We prove that this equilibrium point *is very robust with respect to an increase in the degradation rate*.

Our first result analyzes the effect of increasing d_r on the roots of (16). More precisely, we assume that for some cell C^{ij}, (16) admits a solution $(r_e, c_e) \in C^{ij}$ for $d_r = d_n$, and study how (r_e, c_e) is affected by

Figure 8. Cell transitions in the neighborhood of e^{14}

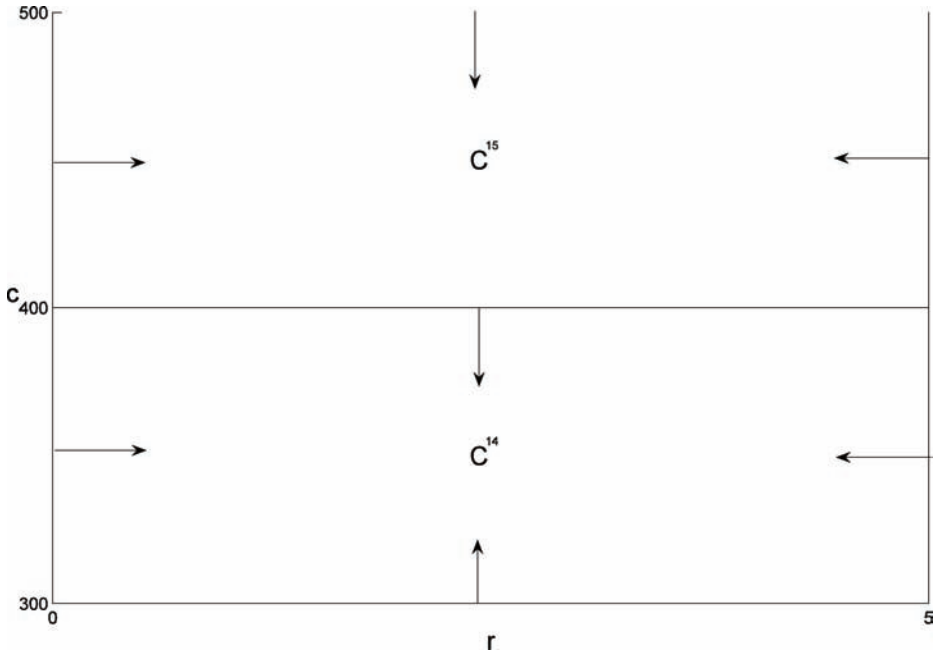


Figure 9. Phase space trajectories. CI degradation rate increased to $4d_n$.

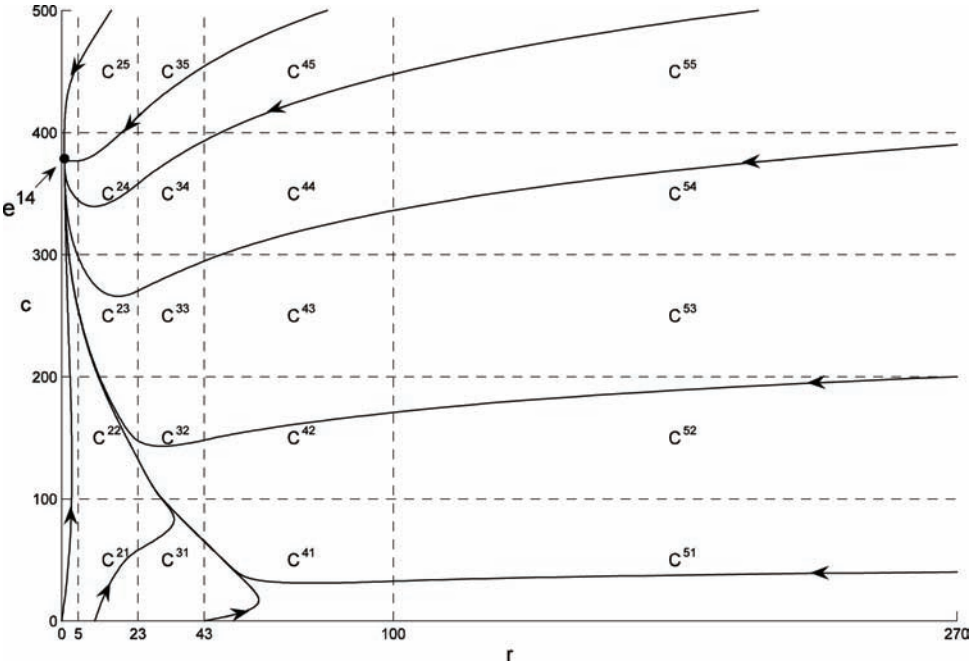
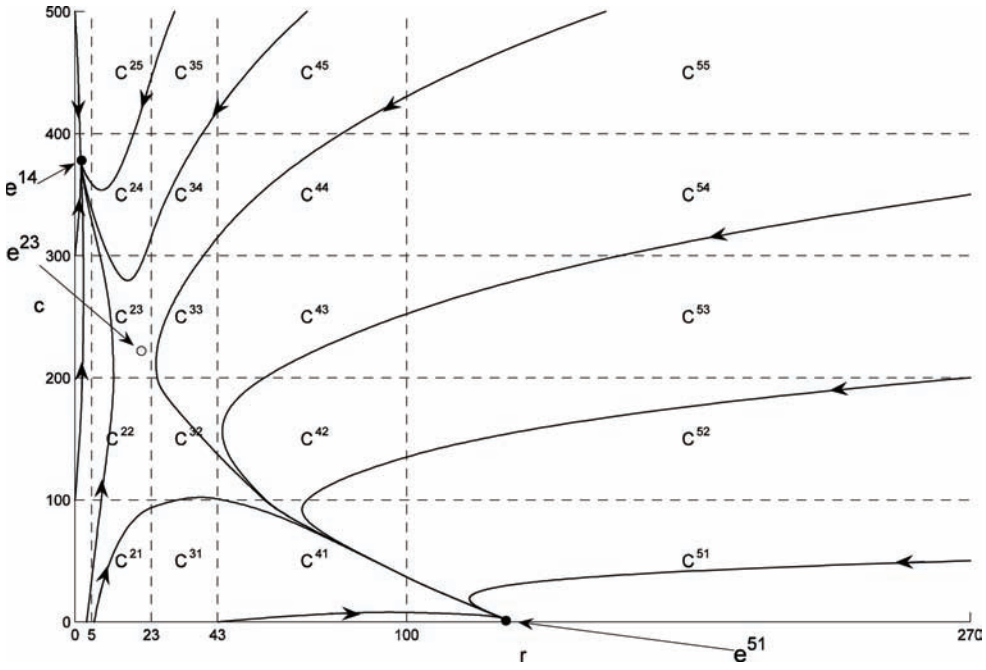


Figure 10. Phase space trajectories. CI degradation rate increased to $2d_n$



increasing the value of the parameter d_r. For the sake of notational convenience, we write p_k instead of p_k^{ij}.

Proposition 7. Consider the equations:

$$\begin{aligned} 0 &= A_r\left(rcp_1 + rp_2 + cp_3 + p_4\right) - rd_r, \\ 0 &= A_c\left(rcp_5 + rp_6 + cp_7 + p_8\right) - cd_c. \end{aligned} \tag{18}$$

Denote $\delta := p_3p_5 - p_1p_7 + p_1d_c/A_c$ Suppose that $\delta A_rA_c \neq 0$ and let: $q_1 = p_5 / (\delta A_r)$, $q_2 = (p_1p_6 - p_2p_5)/\delta_7q_3 = (p_1p_8 - p_4p_5)/\delta_7v = p_1p_2$, $w = p_1q_1$, $n = p_2 + p_3q_2 + p_1q_3$, $m = p_3q_1 - 1/A_r$, $\alpha = v + wd_r$, $\beta = n + md_r$, and $\gamma = p_4 + p_3q_3$. Then the solutions of (18) are:

$$r_{\pm} = \frac{-\beta \pm \sqrt{\Delta}}{2\alpha}, \quad c_{\pm} = (q_1d_r + q_2)r_{\pm} + q_3, \tag{19}$$

where $\Delta = \beta^2 - 4\alpha\gamma$. Suppose that for $d_r = d_n$,

$$r_e = \frac{-\beta - \sqrt{\Delta}}{2\alpha}, \quad c_e = (q_1d_r + q_2)r_e + q_3 \tag{20}$$

is a feasible solution (that is, $(r_e, c_e) \in C^{ij}$). Assume also that $v, w, \gamma > 0$; $p_1, m < 0$; and $\gamma w^2 - nmw + vn^2 > 0$. Then for any $d_r \geq d_n$:

(1) r_e decreases monotonically with d_r, and $\lim_{d_r \to \infty} r_e = 0$.

(2) c_e increases monotonically with d_r, and $\lim_{d_r \to \infty} c_e = q_3 - \frac{w\gamma}{p_1 m}$.

Proof. The two equations in (18) yield $c = (q_1 d_r + q_2)r + q_3$ Substituting this in the first equation of (18) yields $\alpha r^2 + \beta r + \gamma = 0$ and this implies (19). Straightforward algebraic manipulations yield

$$r_e = \frac{-n - md_r - \left|m\right|\sqrt{d_r^2 + s_1 d_r + s_2}}{2(v + wd_r)}, \tag{21}$$

where $s_1 = (2nm - 4w\gamma)/m^2$, and $s_2 = (n^2 - 4v\gamma)/m^2$. Since $v, w > 0$, the denominator in (21) is positive and increases monotonically with d_r. Differentiating the numerator with respect to d_r shows that the numerator is a decreasing function of d_r. Hence, r in (21) is a monotonically decreasing function of d_r. Using (21) and the condition $m < 0$ yields $\lim_{d_r \to \infty} r_e = 0$.

Using (20) and (21) yields $c_e = \frac{-n - md_r - |m|\sqrt{d_r^2 + s_1 d_r + s_2}}{2p_1} + q_3$. We already know that the numerator is a monotonically decreasing function of d_r, and since $p_1 < 0$, c_e is a monotonically increasing function of d_r. Using the fact that $m < 0$, and the expression $\sqrt{1+x} = 1 + x/2 + o(x^2)$ yields

$$\begin{aligned} -md_r - |m|\sqrt{d_r^2 + s_1 d_r + s_2} &= d_r(-m + m\sqrt{1 + s_1/d_r + s_2/d_r^2}) \\ &\approx d_r(-m + m(1 + s_1/(2d_r) + s_2/(2d_r^2))) \\ &= m(s_1/2 + s_2/(2d_r)), \end{aligned}$$

so $\lim_{d_r \to \infty} c_e = \frac{-n + ms_1/2}{2p_1} + q_3$. Using the definition of s_1 completes the proof of Proposition 7.

For cell C^{14}, substituting the parameters in the Appendix yields: $\delta = -6.8E-013$, $q_1 = -13680.9$, $q_2 = -0.0472$, $v = 8.4E-014$, $w = 0.00024$, $n = 4.71E-06$, $m = -0.0861$, $\gamma w^2 - nmw + vm^2 = 1.1E-010$, and $(r_e, c_e)\big|_{d_r = d_n} = (3.8, 376.1)$. Hence, all the conditions of Proposition 7 hold, and as d_r increases, r_e decreases monotonically to 0, and c_e increases monotonically to $q_3 - \frac{w\gamma}{p_1 m} = 379.4$. This implies that for any $d_r \geq d_n$, the solution of (18) satisfies $(r_e, c_e) \in C^{14}$, i.e. *the equilibrium point corresponding to the lytic state remains more or less intact*. This is not so for the two other equilibrium points (see Figures 9 and 10).

The next result shows that the equilibrium point corresponding to the lytic state also maintains its stability.

Proposition 8. Suppose that the conditions of Proposition 7 hold. Let $e(d_r) := (r_e(d_r), c_e(d_r))$ denote the equilibrium point defined in (20). Suppose that for some nominal value $d_r = d_n$, with $d_n > 0$, $e(d_n)$ is LAS. If $p_7, \delta < 0$ and $p_5 > 0$ then $e(d_r)$ is LAS for any $d_r \geq d_n$.

Proof. Denote $y_1 = r - r_e$, $y_2 = c - c_e$. Using (15) and the fact that (r_e, c_e) is a solution of (18) yields:

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} A_r p_1 \\ A_c p_5 \end{pmatrix} y_1 y_2, \tag{22}$$

where $a_{11} = A_r(c_e p_1 + p_2) - d_r$, $a_{12} = A_r(r_e p_1 + p_3)$, $a_{21} = A_c(c_e p_5 + p_6)$, and $a_{22} = A_c(r_e p_5 + p_7) - d_c$. It is well-known that the linear part of (22) will be asymptotically stable if:

$$a_{11} + a_{22} < 0 \tag{23}$$

and

$$a_{11}a_{22} - a_{12}a_{21} > 0. \tag{24}$$

We now show that if (23) and (24) hold for some value $d_r = d_n$, with $d_n > 0$, then they hold for any $d_r \geq d_n$. Since $p_1 < 0$ ($p_5 > 0$), Proposition 7 implies that a_{11} (a_{22}) decreases with d_r. Hence, if (23) holds for $d_r = d_n$, it also holds for any $d_r \geq d_n$.

A straightforward calculation using (20) shows that

$$\frac{a_{11}a_{22} - a_{12}a_{21}}{A_r A_c} = -2\delta c_e + (\frac{d_c}{A_r A_c} - p_7)d_r + \theta,$$

where θ is a constant. Since $p_7, \delta < 0$, Proposition 7 implies that $a_{11}a_{22} - a_{12}a_{21}$ is an increasing function of d_r. Hence, if (24) holds for $d_r = d_n$, it also holds for any $d_r \geq d_n$. This completes the proof of Proposition 8.

Using the parameter values given in the Appendix shows that all the conditions of Proposition 8 hold for cell C^{14}. Summarizing, we suggest the following explanation for the stability puzzle. Under normal conditions (i.e., $d_r = d_n$), the system admits three equilibrium points. Both the lytic and the lysogenic equilibrium points are LAS, and the lysogenic point has a large basin of attraction. This explains the stability of the lysogenic state with respect to perturbations. However, during the SOS response d_r is increased above a certain threshold value, and the dynamic landscape changes dramatically. The equilibrium point corresponding to the lytic state remains more or less intact, while the other two equilibrium points disappear. Thus, all initial states converge to the lytic state.

This explanation is, of course, similar to the one first suggested by (Santillan & Mackey, 2004). However, as demonstrated above, our model is simple enough to allow the study of this bifurcation behavior using rigorous analysis, and not only simulations.

6. DISCUSSION

As noted in (Zhu et al., 2007), biological theories are often of a descriptive nature. In other words, they consist of descriptions and explanations stated in *natural language*. Science can greatly benefit from transforming these verbal descriptions into well-defined mathematical models. Indeed, mathematical models summarize and interpret the empirical data, and are indispensable when we wish to rigorously analyze a dynamic system.

This raises the following problem: how can one convert a given verbal description into a well-defined mathematical model? FM, with its unique ability to handle and manipulate verbal information, constitutes a natural approach for addressing this problem. Application of FM in this context consists of four steps: (1) identifying the state-variables; (2) restating the given verbal descriptions as a set of fuzzy rules relating these variables; (3) defining the fuzzy terms using suitable membership functions; and (4) inferring the fuzzy rule-base to obtain a well-defined mathematical model.

The close connection between the initial verbal description and the resulting mathematical model provides several advantages. The knowledge about the system is represented in three different forms in parallel: (1) the initial verbal descriptions and explanations; (2) the fuzzy rule-base; and (3) the mathematical model obtained by inferring the rules. This provides a synergistic view of the system. For example, simulations and analysis of the mathematical model can be used to check whether the model's behavior is congruent with that actually observed in nature. When this is not the case, it is sometimes possible, due to the If-Then structure of the rules, to determine which fuzzy rule should be altered and how. Inferring the modified rule-base yields a modified mathematical model, and so on. Furthermore, any change in the rule-base can also be interpreted as a change in the initial verbal description, suggesting directions for further research of the original natural phenomenon.

In this chapter, we applied FM to transform a verbal description of the molecular mechanisms underlying the λ switch into a well-defined mathematical model. Simulations indicate that the model provides reasonable qualitative and quantitative fidelity with experimental evidence. Unlike previous models, it is also simple enough to allow a rather detailed analysis. In particular, properties such as the number and location of equilibrium points, and their domains of attraction can be analyzed analytically.

Furthermore, the model provides for the first time a *rigorous* explanation of the so-called stability puzzle of the λ switch. This explanation is similar to the one suggested by (Santillan & Mackey, 2004) based on numerical analysis of bifurcations that appear when the degradation rate is increased. However, the latter model is a fourth-order differential equation with time-delays. Our model is a piecewise-quadratic second-order differential equation. It is thus simple enough to allow a rigorous analysis of the behavior of the equilibrium points for various values of the degradation rate.

REFERENCES

Ackers, G. K., Johnson, A. D., & Shea, M. A. (1982). Quantitative model for gene regulation by λ repressor. *Biophysical Journal, 79*, 1129–1133.

Arkin, A., Ross, J., & McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli cells. Genetics, 149*, 1633–1648.

Aurell, E., & Sneppen, K. (2002). Epigenetics as a first exit problem. *Physical Review Letters, 88*(048101).

Bajec, I. L., Zimic, N., & Mraz, M. (2005). Simulating flocks on the wing: The fuzzy approach. *Journal of Theoretical Biology, 233*, 199–220. doi:10.1016/j.jtbi.2004.10.003

Bakk, A., Metzler, R., & Sneppen, K. (2004). Sensitivity of OR in phage λ. *Biophysical Journal, 86*, 58–66. doi:10.1016/S0006-3495(04)74083-7

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology, 9*, 67–103. doi:10.1089/10665270252833208

de Jong, H., Gouze, J.-L., Hernandez, C., Page, M., Sari, T., & Geiselmann, J. (2004). Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology, 66*, 301–340. doi:10.1016/j.bulm.2003.08.010

Dubois, D., Nguyen, H. T., Prade, H., & Sugeno, M. (1998). Introduction: The real contribution of fuzzy systems. In H. T. Nguyen & M. Sugeno (Eds.), *Fuzzy systems: Modeling and control* (pp. 1-17). Kluwer.

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature, 403*, 339–342. doi:10.1038/35002131

Hawley, D. K., & McClure, W. R. (1982). Mechanism of activation of transcription initiation from the λ PRM promoter. *Journal of Molecular Biology, 157*, 493–525. doi:10.1016/0022-2836(82)90473-9

Kandel, A. (Ed.). (1992). *Fuzzy expert systems*. CRC Press.

Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic: Theory and applications.* Prentice Hall.

Little, J. W., Shepley, D. P., & Wert, D. W. (1999). Robustness of a gene regulatory circuit. *The EMBO Journal, 18*, 4299–4307. doi:10.1093/emboj/18.15.4299

Margaliot, M. (2007). Mathematical modeling of natural phenomena: A fuzzy logic approach. In P. P. Wang, D. Ruan & E. E. Kerre (Eds.), *Fuzzy logic-a spectrum of theoretical and practical issues* (pp. 113-134). Springer.

Margaliot, M. (2008). Biomimicry and fuzzy modeling: A match made in heaven. *IEEE Computational Intelligence Magazine, 3*, 38–48. doi:10.1109/MCI.2008.926602

Margaliot, M., & Langholz, G. (1999). Fuzzy Lyapunov based approach to the design of fuzzy controllers. *Fuzzy Sets and Systems, 106*, 49–59. doi:10.1016/S0165-0114(98)00356-X

Margaliot, M., & Langholz, G. (2000). *New approaches to fuzzy modeling and control-design and analysis.* World Scientific.

McAdams, H. H., & Shapiro, L. (1995). Circuit simulation of genetic networks. *Science, 269*, 650–656. doi:10.1126/science.7624793

Novak, V. (2005). Are fuzzy sets a reasonable tool for modeling vague phenomena? *Fuzzy Sets and Systems, 156*, 341–348. doi:10.1016/j.fss.2005.05.029

Ptashne, M. (1986). *A genetic switch, gene control, and phage lambda*. Cambridge, MA: Cell Press.

Ptashne, M. (1992). *A genetic switch: Phage λ and higher organisms.* Cambridge, MA: Blackwell Scientific Publications and Cell Press.

Ptashne, M. (2004). *A genetic switch* (3rd ed.). Cold Spring Harbor Laboratory Press.

Rashkovsky, I., & Margaliot, M. (2007). Nicholson's blowflies revisited: A fuzzy modeling approach. *Fuzzy Sets and Systems, 158,* 1083–1096. doi:10.1016/j.fss.2006.11.001

Reinitz, J., & Vaisnys, J. R. (1990). Theoretical and experimental analysis of the phage lambda genetic switch implies missing levels of cooperativity. *Journal of Theoretical Biology, 145,* 295–318. doi:10.1016/S0022-5193(05)80111-0

Rozanov, D. V., D'Ari, R., & Sineoky, S. P. (1998). RecA-independent pathways of lambdoid prophage induction in Escherichia coli. *Journal of Bacteriology, 180,* 6306–6315.

Rozin, V., & Margaliot, M. (2007). The fuzzy ant. *IEEE Computational Intelligence Magazine, 2,* 18–28. doi:10.1109/MCI.2007.906684

Santillan, M., & Mackey, M. C. (2004). Why the lysogenic state of phage λ is so stable: A mathematical modeling approach. *Biophysical Journal, 86,* 75–84. doi:10.1016/S0006-3495(04)74085-0

Shea, M. A., & Ackers, G. K. (1985). The OR control system of bacteriophage lambda. A physical-chemical model for gene regulation. *Journal of Molecular Biology, 181,* 211–230. doi:10.1016/0022-2836(85)90086-5

Siler, W., & Buckley, J. J. (2004). *Fuzzy expert systems and fuzzy reasoning*. Wiley-Interscience.

Sousa, J. M. C., & Kaymak, U. (2002). *Fuzzy decision making in modeling and control*. World Scientific.

Torres, A., & Nieto, J. J. (2006). Fuzzy logic in medicine and bioinformatics. *Journal of Biomedicine & Biotechnology,* 1–7. doi:10.1155/JBB/2006/91908

Tron, E., & Margaliot, M. (2004). Mathematical modeling of observed natural behavior: A fuzzy logic approach. *Fuzzy Sets and Systems, 146,* 437–450. doi:10.1016/j.fss.2003.09.005

Tron, E., & Margaliot, M. (2005). How does the Dendrocoleum lacteum orient to light? A fuzzy modeling approach. *Fuzzy Sets and Systems, 155,* 236–251. doi:10.1016/j.fss.2005.03.008

Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM, 37,* 77–84. doi:10.1145/175247.175255

Zadeh, L. A. (1996). Fuzzy logic=computing with words. *IEEE transactions on Fuzzy Systems, 4,* 103–111. doi:10.1109/91.493904

Zhi-fen, Z., Tong-ren, D., Wen-zao, H., & Zhen-xi, D. (1992). *Qualitative theory of differential equations*. American Mathematical Society.

Zhu, X., Yin, L., Hood, L., Galas, D., & Ao, P. (2007). Efficiency, robustness, and stochasticity of gene regulatory networks in systems biology: λ switch as a working example. In S. Choi (Ed.), *Introduction to systems biology.* Humana Press.

KEY TERMS AND DEFINTIONS

Bifurcation: In mathematical models, a bifurcation occurs when a small change made to a parameter value of a system causes a sudden qualitative or topological change in its behavior.

Equilibrium Point: A point $\tilde{x}$ is called an equilibrium point of the differential equation $\dot{x}(t) = f(x(t))$ if $f(\tilde{x}) = 0$.

Stability of an Equilibrium Point: An equilibrium point $\tilde{x}$ is stable if trajectories that start near $\tilde{x}$ are qualitatively similar to the trajectory emanating from the equilibrium point, i.e. the trajectory $x(t) = \tilde{x}$.

Robustness of an Equilibrium Point: An equilibrium point $\tilde{x}$ is robust if it maintains its qualitative properties even when the values of certain parameters change.

Fuzzy Modeling: A systematic approach for transforming verbal descriptions into well-defined mathematical models.

Gene Regulation: The controlled on/off switching of sets of genes according to internal and external conditions.

λ Switch: The genetic mechanism controlling two alternative pathways in the λ virus. Commonly used as a paradigm of gene regulation.

Stability Puzzle: The λ switch demonstrates both: (1) very fast switching; and (2) robustness to random perturbations. These properties are usually contradicting, and their coexistence in the switch is known as the stability puzzle.

ENDNOTES

1 from the Greek, *Lysis*, act of loosening. *Lysogenic*, capable of producing or undergoing lysis.

2 Note that RecA cleaves *CImonomers*, whereas $r(t)$ in our model is the *total* amount of *CI* molecules. In order to obtain a more accurate qualitative value for the bifurcation parameter, a more detailed model of the effect of cleavage on the degradation rate d_r is needed.

APPENDIX

Table 1. Parameters of PWL functions; $\mu_{\underline{medium\_r}}(r) = s(r;\alpha^1,\overline{\beta^3})$, $\mu_{low\_r}(r) = s(r;\alpha^1,\beta^1)$, $\mu_{low\_c}(c) = s(c;\alpha^2,\overline{\beta^4})$, *and* $\mu_{not\_very\_high\_c}(c) = s(c;\alpha^2,\beta^5)$

α^1	β^1	$\overline{\beta^3}$
0	1	0.0001973
5	0.9748	0.02685
23	0.2644	0.7785
43	0.08677	0.9664
100	0.001903	1
1000	0	0.7114

Table 2.

α^2	$\overline{\beta^4}$	$\overline{\beta^5}$
0	1	1
100	0.4815	0.9358
200	0.2346	0.7643
300	0.1604	0.5807
400	0.1156	0.426
1000	0.04012	0

Table 3. Parameter values of the model (15)

A_r	11
A_c	12
d_r	1/2943 sec^{-1}
d_c	1/5194 sec^{-1}
a_1	0.008195 sec^{-1}
a_2	0.00078825 sec^{-1}
a_3	0.0133 sec^{-1}

Table 4. Parameter values p_1, …, p_4 in each cell (see (15))

	p_1	p_2	p_3	p_4
C^{11}	-2.059e-007	3.971e-005	-4.096e-006	0.0007899
C^{12}	-9.804e-008	2.892e-005	-1.95e-006	0.0005753
C^{13}	-2.944e-008	1.52e-005	-5.856e-007	0.0003024
C^{14}	-1.78e-008	1.171e-005	-3.541e-007	0.0002329
C^{15}	-4.995e-009	6.588e-006	-9.937e-008	0.0001311
C^{21}	-1.613e-006	0.0003111	2.941e-006	-0.0005672
C^{22}	-7.682e-007	0.0002266	1.4e-006	-0.0004131
C^{23}	-2.306e-007	0.0001191	4.205e-007	-0.0002171

C^{24}	-1.395e-007	9.175e-005	2.543e-007	-0.0001673
C^{25}	-3.914e-008	5.162e-005	7.135e-008	-9.411e-005
C^{31}	-3.63e-007	7e-005	-2.581e-005	0.004978
C^{32}	-1.728e-007	5.099e-005	-1.229e-005	0.003626
C^{33}	-5.189e-008	2.68e-005	-3.691e-006	0.001906
C^{34}	-3.138e-008	2.064e-005	-2.232e-006	0.001468
C^{35}	-8.806e-009	1.161e-005	-6.263e-007	0.000826
C^{41}	-1.893e-008	3.651e-006	-4.061e-005	0.007831
C^{42}	-9.015e-009	2.659e-006	-1.934e-005	0.005704
C^{43}	-2.707e-009	1.398e-006	-5.806e-006	0.002998
C^{44}	-1.637e-009	1.077e-006	-3.511e-006	0.00231
C^{45}	-4.593e-010	6.058e-007	-9.852e-007	0.001299
C^{51}	1.363e-008	-2.629e-006	-4.386e-005	0.008459
C^{52}	6.492e-009	-1.915e-006	-2.089e-005	0.006162
C^{53}	1.949e-009	-1.007e-006	-6.271e-006	0.003239
C^{54}	1.179e-009	-7.755e-007	-3.792e-006	0.002495
C^{55}	3.308e-010	-4.363e-007	-1.064e-006	0.001404

Table 5. Parameter values p_5, …, p_8 in each cell (see (15))

	p_5	P_6	p_7	p_8
C^{11}	4.301e-008	-6.698e-005	-8.538e-006	0.0133
C^{12}	1.149e-007	-7.417e-005	-2.28e-005	0.01472
C^{13}	1.23e-007	-7.58e-005	-2.442e-005	0.01505
C^{14}	1.036e-007	-6.998e-005	-2.057e-005	0.01389
C^{15}	4.756e-008	-4.756e-005	-9.441e-006	0.009441
C^{21}	3.37e-007	-0.0005248	-1.001e-005	0.01559
C^{22}	9e-007	-0.0005811	-2.673e-005	0.01726
C^{23}	9.639e-007	-0.0005939	-2.863e-005	0.01764
C^{24}	8.118e-007	-0.0005483	-2.411e-005	0.01628
C^{25}	3.726e-007	-0.0003726	-1.107e-005	0.01107
C^{31}	7.582e-008	-0.0001181	-4.001e-006	0.006232
C^{32}	2.025e-007	-0.0001308	-1.069e-005	0.0069
C^{33}	2.169e-007	-0.0001336	-1.144e-005	0.007052
C^{34}	1.827e-007	-0.0001234	-9.639e-006	0.00651
C^{35}	8.384e-008	-8.384e-005	-4.424e-006	0.004424
C^{41}	1.271e-008	-1.98e-005	-1.288e-006	0.002005
C^{42}	3.395e-008	-2.192e-005	-3.438e-006	0.00222
C^{43}	3.636e-008	-2.24e-005	-3.683e-006	0.002269
C^{44}	3.063e-008	-2.068e-005	-3.102e-006	0.002095
C^{45}	1.406e-008	-1.406e-005	-1.424e-006	0.001424
C^{51}	1.805e-011	-2.812e-008	-1.805e-008	2.812e-005
C^{52}	4.821e-011	-3.113e-008	-4.821e-008	3.113e-005

C^{53}	5.164e-011	-3.182e-008	-5.164e-008	3.182e-005
C^{54}	4.349e-011	-2.937e-008	-4.349e-008	2.937e-005
C^{55}	1.996e-011	-1.996e-008	-1.996e-008	1.996e-005

Chapter 25
Petri Nets and GRN Models

Ina Koch
Beuth University for Technology Berlin, Germany; Max Planck Institute for Molecular Genetics, Germany

ABSTRACT

In this chapter, modeling of GRNs using Petri net theory is considered. It aims at providing a conceptual understanding of Petri nets to enable the reader to explore GRNs applying Petri net modeling and analysis techniques. Starting with an overview on modeling biochemical networks using Petri nets, the state-of-the-art with focus on GRNs is described. Other modeling techniques, for example, hybrid Petri nets are discussed. Basic concepts of Petri net theory are introduced involving special analysis techniques for modeling biochemical systems, for example, MCT-sets, T-clusters, and Mauritius maps. To illustrate these Petri net concepts, a more complex case study–the gene regulation in Duchenne Muscular Dystrophy–is explained in detail, considering the biological background and the interpretation of analysis results. Considering both, advantages and disadvantages, the chapter demonstrates the usefulness of Petri net modeling, in particular for GRNs.

INTRODUCTION

In the last years, modern high-throughput technologies enabled scientists to get a huge amount of qualitative and quantitative data on biological processes in the cell. Nearly complete metabolic networks of several organisms are available (Edwards, 2000), (Schilling, 2002). At present, the amount of qualitative data increases much faster than of quantitative data. In particular for GRNs, gene expression at mRNA level can be determined by various experimental high-throughput technologies. Due to experimental limits, the measurement of quantitative data in vitro as well as in vivo is often infeasible. In many cases, qualitative data is the only source for getting information about the system behavior. With the changing

DOI: 10.4018/978-1-60566-685-3.ch025

relation of available qualitative and quantitative data of biochemical systems as one important reason for applying discrete methods, different approaches for qualitative modeling have been developed. These methods range from Boolean methods to stoichiometry based approaches such as elementary mode analysis (Schuster, 1993), extreme pathway analysis (Schilling, 2000), flux coupling analysis (Larhlimi, 2006), and T-invariant analysis (Heiner, 2004).

The inconsistency and incompleteness of data, evoked by difficult and different measuring conditions, desire for modeling approaches which allow for combination of data at different abstraction levels in one model. In particular for biochemical networks, this property plays a crucial role, because, for example, gene regulatory processes are linked to signal transduction processes and/or metabolic processes. To investigate the interactions and dependencies of these different processes as they occur in the cell, we need to model these interactions in a unique description language.

Petri net (PN) theory offers the possibility to model systems at different abstraction levels within one model. Moreover, freely available PN tools often provide an intuitive graphic representation of the system with easily operating editors. This facilitates the communication between experimentally and theoretically working scientists, what is particularly useful in strong interdisciplinary fields like systems biology.

First PN models of biochemical processes have been developed by Reddy et al. (Reddy, 1994; Reddy et al., 1993 and 19961996), modeling the metabolic systems of the fructose metabolism in liver and the combined glycolysis and pentose phosphate pathway in erythrocytes.

In the past 15 years, many different applications of PNs to biochemical systems have been published. Modeling of metabolic networks as PNs is described in (Hofestädt, 1994), (Genrich, 2001), (Voss, 2003), (Oliveira, 2003), and (Koch, 2005). The analysis of signal transduction networks using PNs was introduced by (Lee, 2004), (Takai-Igarashi, 2005), and (Sackmann et al. 2006). For these network types, mostly qualitative discrete simulation and analysis techniques have been applied. The foundations of quantitative PN modeling are described in (Hofestädt & Thelen, 1998) and (Koch & Heiner, 2008).

The transition from Boolean networks to PNs for analyzing gene regulation is developed in (Steggles et al., 2006) and extended in Steggles et al., 2007). Marwan et al. (Marwan et al., 2005) reconstructed a regulatory network, controlling commitment and sporulation in a bacterium.

For modeling gene regulation, often various biochemical systems using different PN types were considered. For example, Goss & Peccaud (Goss & Peccaud, 1998), (Goss & Peccaud, 1999) investigated the genetic network controlling ColE1 plasmid regulation using stochastic PNs. Also hybrid PNs that comprise qualitative as well as quantitative properties into one model have been applied to different biochemical systems (Matsuno et al., 2000) (Chen, 2003), (Hardy, 2004; Matsuno et al., 2003), (Hardy, 2004), (Saito, 2006). The emphasis of these approaches is the analysis via simulation of gene regulation.

Besides investigations focused on these three biological network types, there are publications that combine different abstraction levels into one PN. For example, Simão et al. (Simão et al., 2005) combine gene regulation and metabolic processes focusing on the simulation. The approach considers the qualitative modeling of the biosynthesis of tryptophan in *E.coli*.

Nutsch et al. (Nutsch, 2005) modeled the kinetic mechanism of flagellar motor switching and its sensory control using first a qualitative PN model, which was then refined to give a quantitative one.

Kielbassa et al. (Kielbassa, 2008) developed a PN model which describes the U1 snRNP (uridine rich small nuclear ribonucleoprotein) assembly pathway in alternative splicing in human cells, considering signaling processes, transport processes, and gene expression.

For an overview of PN approaches in biology, see (Hardy, 2004), (Matsuno et al., 2006), (Chaouiya, 2007), and (Koch & Heiner, 2008). A tool comparison of PN software tools to study properties and dynamics of biological systems can be found in (Peleg, 2005).

In this chapter, we emphasize discrete modeling and analysis of gene regulation illustrated by PN modeling of gene regulatory processes concerning Duchenne Muscular Dystrophy (DMD). The chapter is organized as follows. We begin in Section X.2 with a conceptual introduction into PN fundamentals using simple biological examples. We give basic definitions and introduce those analysis techniques, which are useful for modeling GRNs. In Section 3, we explain modeling of gene regulation in crucial processes for DMD, discussing the application of the PN concepts. In Section 4, after a short summary we conclude providing some key points.

PETRI NET FUNDAMENTALS

Petri nets have been introduced by Carl Adam Petri (Petri, 1962) in his dissertation to describe, simulate, and analyze systems of causally related concurrent processes. Many applications, for example, for modeling manufacturing processes (Proth, 1997) or communication networks (Billington, 1999), led to the development of new theorems, algorithms, and tools. Besides qualitative discrete analysis, PN theory was extended by quantitative concepts as in stochastic PNs (Bause, 1996), (Haas, 2002), and continuous PNs (David, 2005). Nowadays, there exist many applications of PN theory to model technical systems, administrative systems, business process management (van der Aalst, 1999), and others (Reisig, 1985). For an introduction on PN theory, see (Peterson, 1981), (Reisig, 1985), (Murata, 1989), (Starke, 1990), and/or (Baumgarten, 1996). A good introduction into continuous and hybrid PNs can be found in (Alla, 1998) and (David, 2005). To get an overview about existing PN methods, literature, and tools, visit the website (TGI-group, 2008).

Petri Net Fundamental Terms and Notations

Petri nets are directed, bipartite, and labeled graphs. They consist of two types of vertices, one for the passive and one for the active system elements. Vertices (or nodes) which describe the passive system elements, $p \in P$, are called *places*, and are visualized by circles. *Transitions*, $t \in T$, describe the active elements of the system. They are drawn as rectangles of different size, e.g., as squares or flat bars. Places and transitions are connected by directed *edges* (or arcs), $f \in F$, in such a way that only vertices of different type are connected. The edges are labeled by positive integer numbers. Movable objects, the *tokens*, can be located on the places. The distribution of tokens over the net represents a certain *marking*, m, and defines a certain *system state* of the net.

Definition (Petri Net)

The 5-tuple $N = (P, T, F, W, m_0)$ is called a Petri net, if it holds:

- P and T are two finite, non-empty sets with $P \cap T = \varnothing$, $P \cup T \neq \varnothing$.
  The elements of the sets P and T are called places and transitions, respectively.

- F is a two digit relation with $F \subseteq (P \times T) \cup (T \times P)$.
  The elements of F are called arcs. F is called the *flux relation* of N.
- W: $F \rightarrow N$, is the weight of the arcs.
- $m_0 : P \rightarrow N_0$, is the initial marking of N.

The directions of the edges define for each transition a set of pre-places and a set of post-places, denoted as ${}^\bullet t := \{p \mid p \in P \wedge (p, t) \in F\}$ and $t^\bullet := \{p \mid p \in P \wedge (t, p) \in F\}$, respectively. Accordingly, we define for each place a set of pre-transitions and a set of post-transitions, denoted as ${}^\bullet p := \{t \mid t \in T \wedge (t, p) \in F\}$ and $p^\bullet := \{t \mid t \in T \wedge (p, t) \in F\}$, respectively. It is possible to model transitions without pre- or post-places and places without pre- or post-transitions. Transitions without pre- or post-places are used to represent the interface with the surroundings of the system, e.g., for substance input and output. We denote transitions without pre-places as *input transitions* and transitions without post-places as *output transitions*. Accordingly, we define *input places* and *output places*.

In biochemical applications, places represent the chemical compounds:

- in metabolic systems, the metabolites (e.g., sucrose, glucose, ATP, ADP);
- in signal transduction systems, the proteins in different states (e.g., in their activated or inactivated form, phosphorylated or dephosphorylated form) or in protein complexes;
- in GRNs, genes at different expression levels (e.g., overexpressed and underexpressed genes), or silencers and enhancers, but also proteins, protein complexes, and complexes between proteins and nucleic acids.

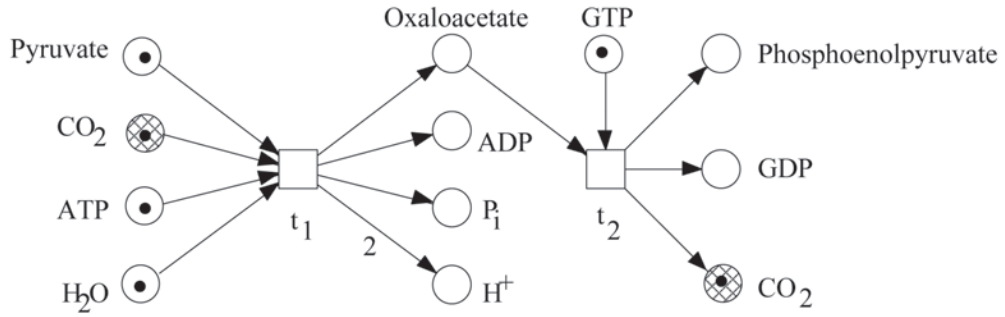
Transitions in biochemical networks represent the chemical reactions:

- in metabolic systems, the enzyme-catalyzed conversions of metabolites, often named after the enzyme (e.g., invertase, hexokinase);
- in signal transduction systems, complex formation and decay reactions (e.g., dimerization, dissociation) or activation/deactivation or phosphorylation/dephosphorylation;
- in GRNs, gene silencing and enhancing processes, transcriptional processes, binding, activation/deactivation, phosphorylation/dephosphorylation, but also complex forming and decay reactions.

For larger systems, different kinds of networks can overlap, because gene expression can influence metabolisms and/or signal transduction pathways and vice versa.

To illustrate the new terms, let us consider the example in Figure 1. It represents a part of the gluconeogenesis pathway, which converts pyruvate into glucose. Following reactions form phosphoenolpyruvate from pyruvate by way of oxaloacetate through the action of pyruvate carboxylase and phosphoenolpyruvate carboxykinase. The corresponding PN consists of two transitions, t_1 and t_2. In metabolic PNs, these transitions are often named after the enzyme which catalyzes the corresponding chemical reactions. Transition, t_1, has four pre-places, pyruvate, CO_2 (carbon dioxide), ATP (adenosine triphosphate), and H_2O (water), and four post-places, oxalacetate, ADP (adenosine diphosphate), P_i (inorganic orthophosphate), and H^+ (hydrogen ion). Transition, t_2, exhibits two pre-places, oxalacetate and GTP (guanosine triphosphate), and three post-places, phosphor-enolpyruvate, GDP (guanosine diphosphate), and CO_2. The node CO_2 is graphically represented by two *logical places* named CO_2, indicating the same node in the underlying graph. Logical places were introduced to achieve a better

Figure 1. A metabolic PN and its chemical stoichiometric equations describing a part of the gluconeogenesis pathway. Places represent the metabolites and transitions the chemical reactions. The arc weights correspond to the stoichiometric factors. Transition, t_1, has four pre-places, pyruvate, CO_2 (carbon dioxide), ATP (adenosine triphosphate), and H_2O (water), and four post-places, oxalacetate, ADP (adenosine diphosphate), P_i (inorganic orthophosphate), and H^+ (proton). Transition, t_2, exhibits two pre-places, oxalacetate, GTP (guanosine triphosphate), and three post-places, phosphoenolpyruvate, GDP (guanosine diphosphate), and CO_2. The node CO_2 is graphically represented by two logical places named CO_2, indicating the same node in the underlying graph.



$$t_1:\ \text{Pyruvate} + CO_2 + \text{ATP} + H_2O \xrightarrow{\text{Pyruvate carboxylase}} \text{oxaloacetate} + \text{ATP} + P_i + 2\,H^+$$

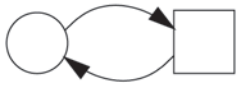
$$t_2:\ \text{Oxaloacetate} + \text{GTP} \xrightarrow{\text{Phosphoenolpyruvate carboxykinase}} \text{phosphoenolpyruvate} + \text{GDP} + CO_2$$

graphical arrangement. Thus, both places always carry the same number of tokens. The arc (t_1, H^+) is labeled by the weight two, whereas all other arcs carry the weight one, not explicitly drawn in graphic depictions. In metabolic systems, the arc weights correspond to the stoichiometric factors of the underlying stoichiometric chemical reaction equation.

PNs can contain loops, if there are two arcs in opposite directions between a place and a transition, see Figure 2. Both arcs can be summarized into one arc, which is called a *read arc* or *test arc*. The place, participating in the read arc, can, e.g., represent a side condition, because a sufficient amount of tokens on this place is necessary to enable the transition, participating in the read arc. Therefore, in chemical reactions, catalysts, e.g., enzymes, can be modeled as a place, participating in a read arc, because tokens on these places are not consumed.

The dynamic properties of a PN are defined by the *firing* of transitions. This corresponds to the occurrence of an action or, in biological context, of a chemical reaction. The firing rule ascertains how the action takes place. Before a transition can fire, it has to be *enabled*. A transition is enabled, if all *pre-conditions* and *post-conditions* are fulfilled. Pre-conditions are represented by pre-places and their markings. If the pre-places carry at least as many tokens as indicated by the weights of the respective outgoing arcs the pre-conditions are fulfilled. Accordingly, post-conditions are defined by post-places and their marking. The post-places have to be able to additionally carry as many tokens as indicated by the weights of the respective incoming arcs. We have not explicitly defined the capacity of a place,

Figure 2. Three equivalent graphical representation for read or test arcs. Read arcs form loops. They can be used to model catalysts, because catalysts, e.g., enzymes, are necessary for the occurrence of a biochemical reaction, but will not be consumed during the reaction.



because we assume that the maximum number of tokens on a place can be infinite. Thus in our models, post-conditions are always fulfilled. We define the minimum number of tokens on the pre-places by the marking

$t^-(p) := W(p, t)$, if $(p, t) \in F$ and

$t^-(p) := 0$, if $(p, t) \notin F$,

and the number of tokens, which are added to each post-place by

$t^+ := W(t, p)$, if $(t, p) \in F$ and

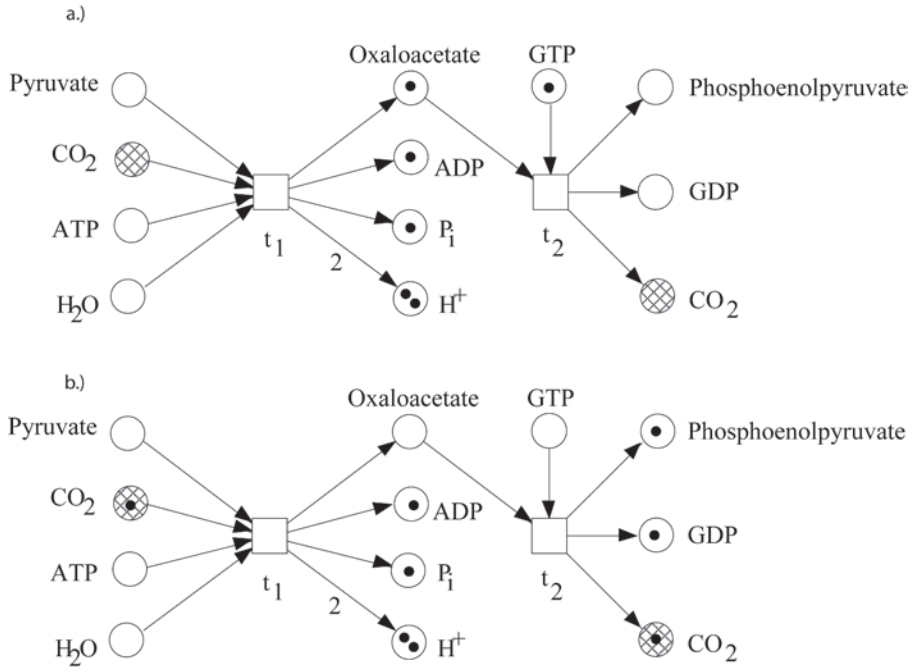
$t^+(p) := 0$, if $(t, p) \notin F$.

Thus, $Dt := t^- - t^+$ gives the change of marking in the considered place.

Definition (Enabled Transition)

Let $N = (P, T, F, W, m_0)$ be a Petri net and m a marking of N. A transition, t, is enabled in m, if $t^- \leq m$.

Figure 3. The firing of the P/T net of Figure 1, whereat t_1 is enabled, because all pre-conditions are fulfilled. a) The PN after firing of transition t_1. As many tokens as indicated by the corresponding arc weights are produced on the post-places (oxaloacetate, ADP, P_i, and H^+). Now, transition t_2 is enabled. b) The PN after firing of transition t_2. On each of the post-places (phosphoenolpyruvate, GDP, and CO_2) one token has been produced, as indicated by the corresponding arc weights.



Definition (Firing Rule)

Let $N = (P, T, F, W, m_0)$ be a Petri net and m a marking of N.

- A transition $t \in T$ can fire in a marking, m, if t is enabled in m.
- After firing of t, a successive marking, m', is reached with $m' := m + Dt$.

For firing of t, we write $m \xrightarrow{t} m'$ for the transition from m to m'. The firing itself is atomic and contains no time relation. PNs that exhibit this timeless firing rule are called place/transition nets (P/T nets). In this chapter, we refer to P/T-nets.

Let us consider again our example of Figure 1. The pre-places of t_1 carry each one token. Because the weights of the arcs between the pre-places and t_1 are all equal to one, all pre-conditions for t_1 are fulfilled, and t_1 is enabled to fire. Transition, t_2, is not enabled, even though if the place GTP carries one token, because one of the pre-conditions represented by the place, oxalacetate, is not fulfilled. After firing of t_1, the PN enters a new system state. Tokens are produced on the post-places of t_1 according to the corresponding arc weights, see Figure 3a. Thus, place H^+ carries two tokens now. Consequently, transition, t_2, is enabled and can fire to give the system state represented in Figure 3b.

The timeless firing rule characterizes the dynamic behavior of P/T nets. The modification of the firing rule by including time intervals for each transition leads to discrete *timed* PNs (Popova-Zeugmann, 2005). Modeling concentrations instead of discrete tokens on places and using concentrations and time dependencies in the firing rule, we define *continuous* PNs. *Hybrid* PNs contain both discrete and continuous places and transitions. If, additionally, probabilities are involved in the firing rule, *stochastic* PNs result.

Definition (Firing Sequence)

Let $N = (P, T, F, W, m_0)$ be a Petri net, m a marking in N, and $s = t_1 \ldots t_n \in T$ a sequence of transitions. The symbol s denotes a *firing sequence*, if a marking, m' in N, exists, such that $m \xrightarrow{t_1 \ldots t_n} m'$ and $m' = m + \sum_{i=1}^{n} \Delta t_i$.

Definition (Reachable Marking)

Let $N = (P, T, F, W, m_0)$ be a Petri net. A marking, m in N, is called *reachable in N* for a marking, m^* in N, if a firing sequence, s, from m^* to m in N exists. If $m^* = m$, we say m is reachable in N.

$R_N(m) := \{m' | m \xrightarrow{*} m'\}$ denotes the set of all *reachable markings* in m. This set is of special interest, because it comprises all possible events defining all system states, and, thus, representing the *state space* of a PN.

Definition (State Space)

Let $N = (P, T, F, W, m_0)$ be a Petri net. The set $R_N(m) := R_N(m_0)$ is called the *system state* of N.

The state space is summarized in the reachability graph, *RG*. The vertices of *RG* represent the different states, and the edges describe state transformations labeled by the responsible transition.

Definition (Reachability Graph, RG)

Let $N = (P, T, F, W, m_0)$ be a Petri net. We call the graph a *reachability graph*, *RG* of N, if:

- The set RN represents the vertices of the graph.
- (m, m') denotes an edge of the graph, if a transition, t, exists with $m \xrightarrow{t} m'$.

The state space gives also insights into non-reachability of certain states and, thus, information about actions that never will take place. For exploration of the system space, the state space should be finite. A system state is finite, if no place exists, carrying an infinite number of tokens. This is expressed by the following property.

Definition (Boundedness)

Let $N = (P, T, F, W, m_0)$ be a Petri net. We call a Petri net N to be *bounded*, if the set of reachable markings is finite.

- A place is *k-bounded*, if there exists a positive integer number, k, which represents the maximal number of tokens on that place.
- A Petri net N is k-bounded, if all its places are k-bounded.
- A Petri net is *structurally bounded*, if it is bounded in every initial marking.

Biochemical PNs are usually modeled as *open* systems, using input and output transitions. Thus, tokens can always enter or leave the model, leading in most cases to places with infinite number of tokens, and thus, to unbounded networks. There are approaches to convert an unbounded network into a bounded one (Koch & Heiner, 2008).

Another important property refers to the *liveness* of a transition and of the entire PN.

Definition (Liveness)

Let $N = (P, T, F, W, m_0)$ be a Petri net, m an arbitrary marking of N, and $t \in T$ an arbitrary transition.

- A transition, t, is *live* in the marking, m of N, if for every marking, $m' \in R_N(m)$, a further marking, $m'' \in R_N(m')$, with $m'' \xrightarrow{t} m'$ exists.
- A transition t is *dead* in the marking, m of N, if for every marking, $m' \in R_N(m)$, it holds: $t^- \not\leq m'$.
- A marking, m, is called *live in N*, if all transitions, $t \in T$, are live in m.
- A marking, m, is called *dead in N*, if all transitions, $t \in T$, are dead in m.
- A transition t is called *live* (*dead*) *in N*, if t is live (dead) in m_0.
- A Petri net, N, is called *live* (*dead*), if m_0 is live (dead) in N.
- A Petri net, N, is called *deadlock-free*, if there is no reachable marking, where all transitions, $t \in T$ in N, are dead.

Biochemical systems should be live, because it is assumed that biological processes can repeatedly occur. A dead system means that none reaction can take place, and the metabolism or signal transduction stops. For modeling drug dependencies, a dead transition or even a dead PN can be desired as effect of the drug after entering the system.

Linear Invariant Analysis

Dynamically defined network properties can often be characterized by linear algebraic methods. In this context, the invariant properties play a crucial role. These invariants are important to characterize the dynamic system behavior, in particular for biochemical systems. Invariant properties are valid in each system state, independently of the current state represented by the current marking. The definition of invariants is based on the incidence matrix. Thus, it is not necessary to generate the whole state space, which often leads to a state space explosion (Valmari, 1998).

Figure 4. The incidence matrix of the PN depicted in Figure 1. Each matrix element contains the number of tokens on the places (written vertically), which will be produced (positive numbers) or consumed (negative numbers) by firing of the transitions (written horizontally). For example the place, ADP, will get one token, when t_1 fires; CO_2 will deliver one token, when t_1 fires, and get a token by firing of t_2.

	t1	t2
ADP	+1	0
ATP	−1	0
CO_2	−1	+1
GDP	0	+1
GTP	0	−1
H^+	+2	0
H_2O	−1	0
Oxalacetate	+1	−1
P_i	+1	0
Phosphoenolpyruvate	0	+1
Pyruvate	−1	0

Definition (Incidence Matrix)

Let $N = (P, T, F, W, m_0)$ be a Petri net. The corresponding *incidence matrix C* is defined by " $1 \leq i \leq m$, $1 \leq j \leq n$:

$$C_{ij} := \begin{cases} W(t_j, p_i), & if (t_j, p_i) \in F \setminus F^{-1} \\ -W(p_i, t_j), & if (p_i, t_j) \in F \setminus F^{-1} \\ W(t_j, p_i) - W(p_i, t_j), & if (t_j, p_i) \in F \cap F^{-1} \\ 0, & otherwise. \end{cases}$$

A matrix element, C_{ij}, denotes the change of the token number on place, p_i, by the firing of the transition, t_j. PNs without loops are defined one-to-one by the incidence matrix. In chemistry, the incidence matrix is known as *stoichiometric matrix*. The incidence matrix of our example of Figure 1 is depicted in Figure 4, where, for example, the place, H^+, will get two tokens, when t_1 fires; CO_2 will deliver one token, when t_1 fires, and get a token by firing of t_2.

Definition (Place Invariant – P-invariant)

Let $N = (P, T, F, W, m_0)$ be a Petri net and C the corresponding incidence matrix. Each non-trivial solution $x \in N^{|T|}$ of the homogeneous equation system $C^T \times x = 0$ is called a *place invariant* (*P-invariant*) of N.

Place-invariants describe rules of token conservation. A P-invariant is a set of places, for which the weighted sum of tokens is always constant, independently from any firing. If $x = (x_1, x_2, \ldots, x_{|P|})$ is a P-invariant in a Petri net N, it holds for each reachable marking, m:

$$\sum_{i=1}^{|P|} x_i \cdot m(p_i) = const. = \sum_{i=1}^{|P|} x_i \cdot m_0(p_i) .$$

In biochemical systems, P-invariants describe conservation rules of compounds. In metabolic systems, ATP and ADP often form a P-invariant. In signal transduction systems, and GRNs those places, which describe the activated and non-activated state of a protein, often build a P-invariant.

Definition (Transition Invariant – T-invariant)

Let $N = (P, T, F, W, m_0)$ be a Petri net and C the corresponding incidence matrix.

- Each non-trivial solution $y \hat{\mathrm{I}} N^{|T|}$ of the homogeneous equation system $C \times y = 0$ is called a *transition invariant* (*T-invariant*) of N.
- Each *Parikh vector* of a firing sequence, which represents a T-invariant, is called a *feasible T-invariant.*

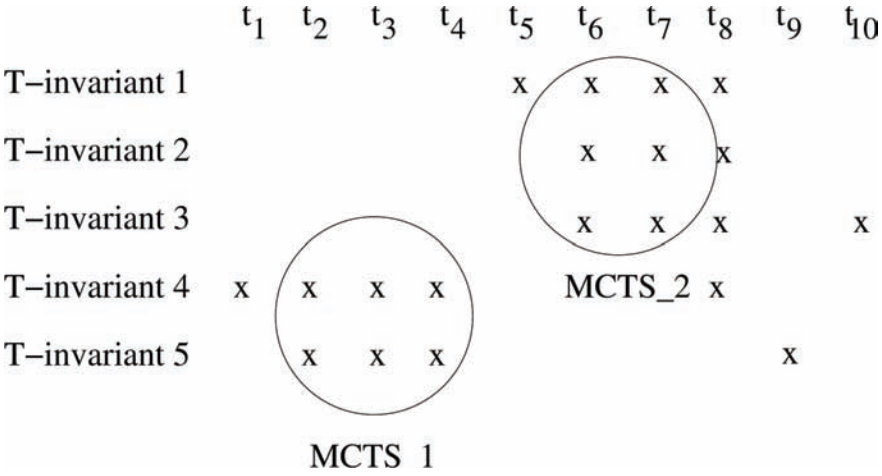
The transitions of a T-invariant and the places in between describe subnets, which are connected and can overlap. Parikh vectors contain frequencies of their elements (for the formal definition see (Parikh, 1966)). In PN theory, T-invariants are Parikh vectors, because they contain the frequencies of firing for each transition. If all transitions of a T-invariant fire according to their frequencies as indicated in the Parikh vector, an arbitrary initial system state will be reached again.

The set of the non-zero entries of an invariant, x, is called the *support* of x, written as *supp(x)*. Because the solution space of these equations is infinite, we are interested in a minimal solution, from which each other solution can be obtained by linear combinations. An invariant x (P- or T-invariant) is *minimal*, if it does not contain any other invariant z, i.e., $\nexists\, z$: supp(z) $\subseteq$ supp*(x)*, and the greatest common divisor of all entries of x is equal to one. In this chapter, we consider minimal invariants, writing *P-invariant* and *T-invariant*, respectively.

We can formulate minimal validation criteria for a biochemical PN (Heiner, 2004; Koch & Heiner, 2008):

1. The net should be connected.
2. All transitions should be a member of at least one T-invariant. Otherwise, this transition does not influence the system behavior. That means that we can remove this transition without any change in the system behavior. If each transition of a PN belongs to at least one T-invariant, the PN is *covered by T-invariants* (*CTI*). Accordingly, we can define the property, *CPI (covered by P-invariants)*, if each place belongs to at least one P-invariant.
3. Each T-invariant should represent a biologically possible pathway, because T-invariants describe the complete basic system behavior. The process of assigning a biological meaning can be complicated and time-consuming for a huge number of T-invariants.

Figure 5. A set of five T-invariants and their transitions. There are two MCT-sets, MCTS1= {t_2, t_3, t_4} and MCTS2 = {t_6, t_7}. Transition, t_2, is not member of MCTS2, because it also occurs in T-invariant4, in which transitions, t_6 and t_7, do not appear.



Extended Invariant Analysis

Because of the complexity and dimension of biochemical PN models, the number of T-invariants can become very large, some hundreds or thousands and even more. To facilitate the examination of a huge amount of T-invariants, we introduce two concepts, *MCT-sets* and *T-clusters*. MCT-sets, first defined in (Sackmann et al., 2005) summarize equal parts of T-invariants.

Definition (Maximal Common Transition Set – MCT-set or MCTS)

Let $N = (P, T, F, W, m_0)$ be a Petri net and X the set of all T-invariants, x. A transition set $A \in T$ is called a *maximal common transition set, MCT-set* or *MCTS*, if and only if $\forall\, x \in X$: $A \subseteq \text{supp}(x) \vee A \cap \text{supp}(x) = \varnothing$.

That means that for all $i, j \in \{1, 2, \ldots, m\}$ two transitions, t_i and t_j, are grouped into the same MCT-set, if and only if they participate in exclusively the same minimal T-invariants, i.e., for all T-invariants, x, it holds: $c_{\{0\}}(x_i) = c_{\{0\}}(x_j)$, where $c_{\{0\}}$ denotes the characteristic function, binary indicating whether the argument is equal to zero or not. Figure 5 depicts an example for two MCT-sets.

MCT-sets and the places in between describe disjoint subnets, which can, in turn, be disconnected. We interpret MCT-sets as building blocks of biochemical PNs. These building blocks can be used for network reduction. Additionally, it can be assumed that the genes involved in transitions of one MCTS are coherently up- and down-regulated, because they operate always together.

MCTS have been used in several applications of PNs to biology (Grafahrend-Belau et al., 2008), (Sackmann et al., 2006), (Grunwald, 2008), (Sackmann et al., 2007), (Koch & Heiner, 2008).

The transitions of an MCT-set, occur always together in a pathway described by the corresponding T-invariant. Thus, performing exhaustive knockout experiments, it is sufficient to consider the knockout of only one transition of an MCT-set.

Whereas T-invariants decompose the network into connected overlapping sub-networks, the concept of MCT-sets leads to decomposition into disjoint sub-networks, which must not necessarily be connected. These sub-networks should reflect a biologically possible behavior. Thus, they can be used for validating the model.

The clustering of T-invariants represents another possibility for network decomposition. The arising *transition clusters* (*T-clusters)*, also combine transitions like MCT-sets, but the classification criteria are not so strong such that the resulting sub-networks can overlap as T-invariants. T-clusters are also used to validate models of large and complex networks, exhibiting a huge amount of T-invariants.

Cluster analysis of a set of objects can be seen as a three step process: (1) selection of a distance measure to compute the distances between all pairs of objects, (2) selection of a clustering algorithm to group the objects based on their distances, and (3) selection of a cluster validity measure to identify the optimal number of clusters for interpretation.

1. The Tanimoto coefficient (Backhaus, 2003) is suitable as distance measure. It defines the similarity between two objects, *i* and *j*, by

$$s_{ij} = s(t_i, t_j) = \frac{a}{a + b + c},$$

where *a* is the number of features present in both objects, *b* is the number of features only present in object, *i*, and *c* is number of features only present in object, *j* . The distance between the two objects, d_{ij}, is then defined by $d_{ij} = 1 - s_{ij}$ (Steinhausen, 1977). In our case, objects are represented by the support vectors of T-invariants.

2. For clustering, we use UPGMA or the Nearest Neighbor approach (Saitou, 1987), (Studier, 1988). For large networks, UPGMA is much faster.
3. The decision, which number of clusters is the correct one or where to cut the cluster tree, respectively, is one of the fundamental problems in unsupervised classification. After the investigation of different indices, we use the Silhouette Width (Rousseeuw, 1987) as default measure.

T-clusters have been applied in several applications (Grunwald, 2008), (Sackmann et al., 2007), (Koch & Heiner, 2008). For a more detailed description with various biochemical examples see (Grafahrend-Belau et al., 2008).

Mauritius Map Analysis

Both, experimental and theoretical knockout analysis, are common approaches to investigate the system behavior of biochemical networks. It is of general interest to know which parts of the network will be affected by the knockout of a certain gene, i.e., a certain reaction. It can also be to asked, which reactions should be knocked out to achieve a desired system behavior.

Once we have validated our PN model according to (Heiner, 2004), (Koch & Heiner, 2008), we use the T-invariants for knockout analysis, because they describe the complete basic system behavior. Knocking out one transition or a set of transitions, we can compute the T-invariants of the modified system.

Comparing the resulting T-invariants with the set of T-invariants of the original system, we identify those regions of the network, which are affected or not affected by the knockout.

To facilitate an exhaustive knockout analysis we use *Mauritius maps*, which represent a new data structure that graphically visualizes the dependencies of T-invariants in terms of a binary tree. Mauritius maps describe dependencies of sub-pathways, which result from T-invariant analysis. Mauritius maps enable for performing a systematic analysis of *in silico* knockout experiments.

Definition (Mauritius Map)

Let $N = (P, T, F, W, m_0)$ be a Petri net and X the set of all T-invariants, x. A finite binary tree, $T = (V, E)$, is called *Mauritius map*, if

- The set V is a finite set of all transitions, x, belonging to at least one T-invariant. The root vertex is located in the lower left corner.
- The set $E = (H, R)$ is a finite set of edges between vertices, indicating dependencies of T-invariants.
- The set H represents horizontal edges, which connect vertices belonging to the same T-invariant.
- The set R represents vertical edges, which connect vertices of the left sub-tree with vertices of the right upper sub-tree belonging to the same T-invariant.

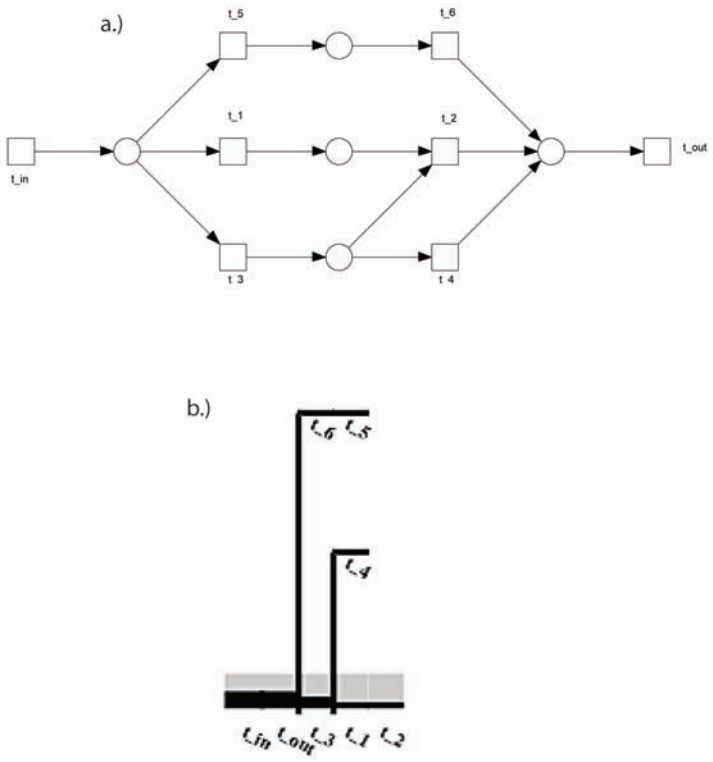
The root vertex has no left sub-tree, but a right sub-tree. This right sub-tree contains all transitions covering the PN. A branch in the tree indicates another T-invariant. Leafs are transitions that form a right sub-tree, consisting only of that transition, and exhibit a left sub-tree. Interior vertices have left and right sub-trees each describing sub-networks. These sub-networks are defined by the vertices following the path from the root to the interior vertex considered. Horizontal and vertical lines, respectively, represent edges to the children, whereas edges to the parent vertices are formed by junction of two such lines. Edges can contain several transitions, forming a set of transitions.

To answer the question for transitions with highest or lowest impact on the net behavior we measure the number of affected, i.e., destroyed, T-invariants. Beginning with the root, the *most important transition* is given by the horizontal line. Knocking out this transition, the impact on the system will be the highest in comparison to knocking out other transitions. Obviously, a transition, which is contained in all T-invariants, represents such a most important transition.

Knocking out a transition or a set of transitions, usually one part of the net remains active, and another part looses its biological function. The corresponding right child and its successors cover all affected pathways. The knockout of a transition fragments the net into two subnets. One subnet (left child) does not contain the transition knocked out, and represents the function of the model not affected by the knockout. The second subnet (right child) depends on the presence of the transition knocked out and will become inactive. Thus, only those pathways, which cover the left child and its successors, are not affected maintaining their biological functionality.

Let us consider the example in Figure 6a. It depicts a small PN, representing signal transduction. The signal has to pass in some way from transition *t_in* to transition *t_out* via transitions *t_1*, *t_2*, *t_3*, *t_4*, and/or *t_5*. The behavior is described by three T-invariants *Inv1 = {t_in, t_1, t_2, t_3, t_out}*, *Inv2 = {t_in, t_3, t_4, t_out}*, and *Inv3 = {t_in, t_5, t_6, t_out}*. Figure 6b depicts the corresponding Mauritius map. The root has no left child, because a knockout of the transition, *t_in* or *t_out*, would be lethal for

Figure 6. A small PN representing a signaling pathway (a) and its Mauritius map (b). a) The signal is going from t_in to t_out. Three T-invariants occur, Inv1 = {t_in, t_1, t_2, t_3, t_out}, Inv2 = {t_in, t_3, t_4, t_out}, and Inv3 = {t_in, t_5, t_6, t_out}. One MCT-set, consisting of t_in and t_out, exists. b) Knocking out transition t_in or t_out destroys the functionality of the system. Knocking out transition, t_3, affects not the left sub-tree (Inv3), but the right sub-tree (Inv1, Inv2). A knocking out of t_1 concerns only Inv1. Edges drawn as rather thick black lines cover large parts of the net, whereas edges drawn as thin gray lines describe an only local influence.

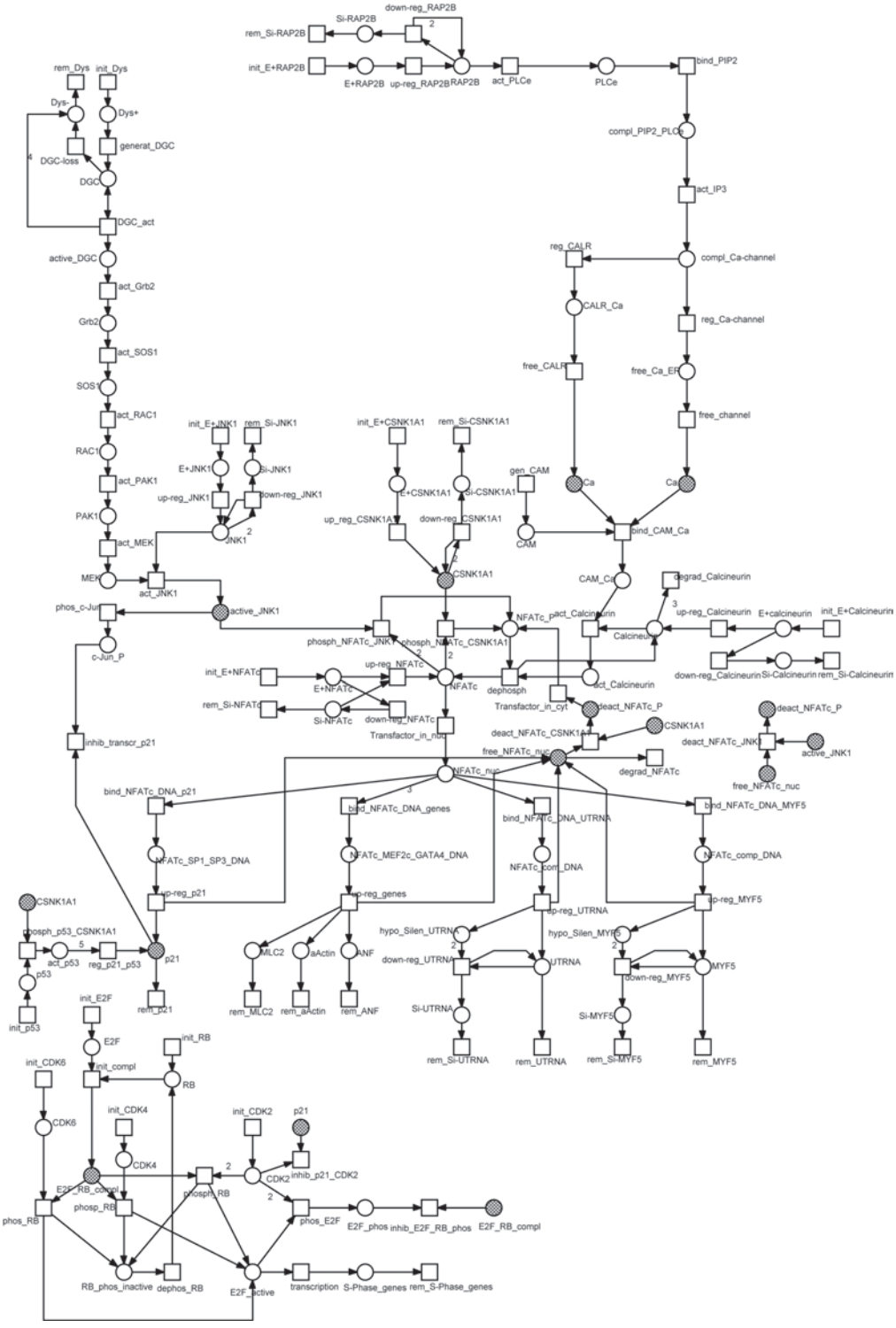


the system. Note that the transitions, *t_in* and *t_out*, represent an MCT-set. Their order is not relevant. A knockout of transition, *t_3*, does not affect the left sub-tree, i.e., the T-invariant, *Inv3*, containing *t_5* and *t_6*, but destroys the functionality defined by the right sub-tree, containing transitions *t_1*, *t_2*, and *t_4*. Knocking out transition *t_1*, *Inv2* is still working, but not transition *t_2*.

MODELING GENE REGULATION IN DUCHENNE MUSCULAR DYSTROPHY

Gene regulation has often been modeled using Boolean approaches (Thomas, 1995). We consider gene regulation as a process that covers more than off- and on-switching of genes. The facility to work at different abstraction levels represents a strength of PN modeling. Gene regulatory processes, which are often induced by signal transduction, can influence, for example, metabolic processes, but also other signaling pathways and vice versa. We are convinced of the necessity to take all these processes into consideration.

Figure 7. The entire PN model of the pathomechanisms of DMD. The black places indicate logical places



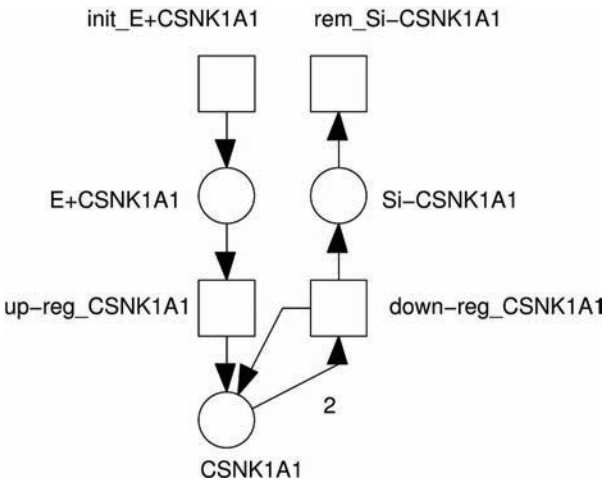
Now, let us consider our example – the processes that influence a disease – Duchenne Muscular Dystrophy (DMD). This disease is still lethal, because no efficient therapy exists. To get a better understanding of the underlying molecular processes and, thus, of new therapeutic ideas, we developed a model of processes downstream the dystrophin gene. Our example is large enough to explain some important aspects of modeling GRNs using PNs. The model is based on own experimental data and has been used by experimentalists to get new insights into system behavior.

Biological Background

DMD is one of the most common inherited human neuromuscular diseases. Primarily boys in early childhood come down with DMD, where all voluntary muscles are affected until death by respiratory and/or cardiac failure.

A mutation in the dystrophin gene, followed by the absence or functional impairment of a subsarcolemmal cytoskeletal protein, causes the disorder. In DMD patients, the protein, dystrophin, is absent. The pathomechanism of DMD, represented by the network downstream of dystrophin, comprises various gene regulatory processes, such as the dystrophin-glycoprotein-complex (DGC) downstream pathway. DGC is formed in presence of dystrophin, and enters a reaction cascade that finally phosphorylates, i.e., deactivates, the nuclear transcription factor of activated T-cells, NFATc. NFATc can be dephosphorylated, i.e., activated, by the protein, calcineurin, which, in turn, is positively regulated by the RAP2B-calcineurin cascade, the RAP2B (Ras related protein 2B) downstream pathway. Activated NFATc can enter the nucleus, where it acts as a transcription factor for different essential genes, such as MYF5 (myogenic factor 5), UTRNA (utrophin A), and p21. The protein, p21, which inhibits the cyclin-dependent kinases (CDK), is, in turn, a negative regulator of the G1 to S progression in the cell cycle.

Figure 8. The sub-net describing the up- and down regulation of the CSNK1A1 gene. The gene is represented at different expression states, the protein itself (CSNK1A1), the enhanced gene (E+CSNK1A1), and the silenced gene (Si-CSNK1A1). The protein CSNK1A1 is produced by the transition, init_E_CSNK1A1, through up-regulation (up-reg_CSNK1A1). The down-regulation is enabled when the place for the protein carries at least two tokens, and removes CSNK1A1 through the transition, down-reg_CSNK1A1, and produces CSNK1A1 again.



That means that p21 slows down cell proliferation, enhancing the dystrophic process. The reduction of p21 and, consequently, the improvement of proliferation of primary myoblasts of DMD patients could be a possible therapeutic approach. Calcineurin could be reduced to deactivate NFATc and, therefore, to decrease the p21 expression (Endesfelder et al., 2003), (Endesfelder et al., 2005).

The Petri Net Model

The whole model depicted in Figure 7 is described in detail in (Grunwald, 2008). The entire PN models, the tables, and analysis files are given in the supplementary material (Koch, 2008a).

The PN is constructed according to the outline of the biological processes and the general PN modeling techniques for GRNs as explained in section X.2.1. The model is mainly based on own experimental data and additional literature knowledge. To fill gaps in the current knowledge, we had to introduce hypotheses, which could not be found in the literature, because they cannot be experimentally tested at the moment. For example, we introduce the transitions *deact_NFATc_CSNK1A1* and *deact_NFATc_JNK1* to model the inactivation of NFATc by CSNK1A1 and by JNK1 in the nucleus. To represent the initiation of silencing processes after an up-regulation of the transcription of the corresponding protein, we introduce the places, *hypo_Silen_UTRNA* and *hypo_Silen_MYF5*.

Modeling GRNs, we are working at a more abstract level of description than in case of metabolic systems. The term of stoichiometry has not that meaning as in metabolic systems, where the stoichiometry is clearly given by knowledge of the proportions of substances taking place in the reaction. Thus, we have no clear definition of arc weights in GRNs. We start with arc weights equal to one. For those reactions, for which we know the proportions from experimental data, we alter the arc weights according to the experimental results. To indicate the delay of down regulation, we increase the incoming arc weight of a transition such that the pre-condition, i.e., the pre-place, has to carry more tokens to enable the transition for firing. This is the case for the transitions *down_reg_UTRNA*, *down_reg_MYF5*, *down-reg_RAP2B*, *down-reg_JNK1*, *down-reg_CSNK1A1*, and *bind_NFATc_DNA_genes*. In Figure 8, for example, the transition, *down-reg_CSNK1A1*, is enabled, if there are at least two tokens on place, *CSNK1A1*. Thus, the transition, *down-reg_CSNK1A1*, is delayed compared to the transition, *up-reg_CSNK1A1*.

We model those genes, whose mRNA expression we obtained experimentally, providing three places. One place stands for up-regulation (enhancing) and the other for down-regulation (silencing). This concerns the genes RAP2B, CSNK1A1, JNK1, NFATc, calcineurin, UTRNA, and MYF5. Additionally, we introduce a place for the gene product. In these cases, we use a special subnet, which is depicted in Figure 8.

Three places represent the different expression states of the CSNK1A gene, the protein itself (*CSNK1A1*), the enhanced CSNK1A1 (*E+CSNK1A1*), and the silenced CSNK1A1 (*Si-CSNK1A1*). The protein CSNK1A1 is produced by the transition, *init_E_CSNK1A1*, through up-regulation (*up-reg_CSNK1A1*). The down-regulation of CSNK1A1 takes place, if its place carries at least two tokens as indicated by the corresponding arc weight. This down-regulation removes CSNK1A1 through the transition, *down-reg_CSNK1A1*, and produces CSNK1A1 again. Note that the structure of the net does not avoid a situation, in which the two places, *E+CSNK1A1* and *Si-CSNK1A1*, will get tokens. Except for some proteins expressed in developmental stages only, the regulation of most genes, including CSNK1A1, does not necessarily result in a complete shutdown and turning-on, respectively. For example, enhancing CSNK1A1 leads to an increased expression by positive transcription factors, comprising a low effect of negative regulators (*Si-CSNK1A1*) as well as the other way around.

For calcineurin and NFATc, an equilibrium between up- and down-regulation appears, because no differences in the mRNA expression levels in patients compared to normal control could be detected. We express this by initiation, up- and down-regulation, and removal of both substances.

We also increase the arc weights to reflect experimental results in cases of protein interaction, see the transitions *DGC_act*, *phos_NFATc_CSNK1A1*, *phos_NFATc_JNK1*, *phosph_RB*, and *phos_E2F*, and to avoid token accumulation for the transitions, *reg_p21_p53* and *degrad_Calcineurin*.

Models using inhibitory arcs exhibit a decreased analysis power. It is well-known that the reachabilty problem, i.e., the question whether a certain marking, m, is reachable from an initial marking, m_0. for PNs with at least two inhibitor arcs is not decidable (Hack, 1976), (Esparza, 1994). To maintain the analysis power we model inhibition by transitions, which remove tokens from the system. For example, p21 inhibits the kinase CDK2, whereas the dissociation of the E2F-RB-complex is inhibited by E2F phosphorylated by CDK2.

Analysis Results

The aim of the analysis is to get insights into the dynamic system behavior by the computation of static and dynamic properties of the system. In this context, we are interested in the computation of general properties, which always hold such as invariant properties. Before *playing* with the model, e.g., by constructing different scenarios, we have to validate the model to get trust into it.

First of all, let us phrase, which questions we want to address:

1. Which properties the net holds?
2. Which basic pathways exist?
3. Do these pathways reflect the main biological behavior?
4. Is the model consistent?
5. Can we reduce the network?
6. Which part of the net is the most and less important one, respectively?
7. Can we easily perform knockout experiments?

To address the first question, we use the INA-tool ("INA - The Integrated Net Analyzer,") producing the output file, duchenne.ina, compare (Koch, 2008b). The results show that our PN model is

- *not ordinary*, i.e., the arc weight of each arc is not equal to one,
- *not homogeneous*, i.e., all outgoing arcs of each place have not the same arc weights,
- *not pure*, i.e., the net contains loops,
- *not conservative*, i.e., the total number of tokens does change, because there are transitions, which add another amount of tokens to their post-places as they consume from their pre-places,
- *not statically conflict-free*, i.e., there are transitions with a common pre-place such that they are in static conflict about the tokens on this pre-place,
- *connected*, i.e., for each node exists a path via undirected edges to each other node,
- *not CPI*, i.e., the net is not covered by P-invariants,
- *CTI*, i.e., the net is covered by T-invariants, and
- *unbounded* and *not structurally bounded*, i.e., there are places, which can get an infinite number of tokens.

- There are transitions without pre-places and transitions without post-places.
- There is *no dead state* reachable.
- The net is *live.*

To answer question 2, we compute the system's invariants and assign a biological meaning to each T-invariant. The net exhibits 107 T-invariants, see the file, duchenne.inv, in the supplementary material (Koch, 2008a), where T-invariants are represented also graphically, using different colors in separate files. The smallest T-invariants each contain three transitions, e.g., T-invariant, *Inv5*, includes transitions *init_E_Calcineurin*, *up-reg_Calcineurin*, and *degrad_Calcineurin*. The T-invariants, *Inv96*, *Inv97*, *Inv98*, and *Inv99*, represent the four largest T-invariants, each containing 32 transitions. All four T-invariants start with the DGC down-stream pathway that activates up-regulated JNK1, and leads to activation of NFATc by dephosporylation through the RAP2B-calicneurin pathway. Then, NFATc migrates into the nucleus to mediate transcription of MLC2, aActin, and ANF. In the nucleus, NFATc can be phosphorylated. i.e., inactivated by JNK1.

The following transitions occur in more than 67% of all T-invariants, thus, they have a crucial meaning for the network behavior, *bind_PIP2*, *act_IP3*, *bind_CAM_Ca*, *gen_CAM*, *act_Calcineurin*, *dephosph*, *act_PLCe*, *Transfactor_in_nuc*, and *Transfac_in_cyt*.

One T-invariant, *Inv55*, exhibits four output transitions, but no input transition. It forms a cyclic pathway, describing the balance between dephosphorylation of NFATc by calcineurin, activated by the RAP2B downstream pathway, and phosphorylation by the kinase CSNK1A1. Consequently, NFATc is not able to migrate into the nucleus and to act as a transcription factor. Thus, the cell down-regulates subsequent gene transcription without the need of protein degradation or gene silencing. All the other T-invariants contain input and output transitions.

Our net model does not exhibit P-invariants, because we did not explicitly modeled substances like ATP, ADP, or AMP, which are always available in one form in the cell and whose amount is generally conserved.

To answer question 3, based on the 107 T-invariants we yield 25 MCT-sets, which consist at least two transitions, see Table 1. The MCT-sets are provided in file, duchenne.mct, in the supplementary material (Koch, 2008a).

Except *M1* and *M22*, the other 23 MCT-sets describe connected sub-networks. The MCT-set *M22* depicted in Figure 9 consists of the transitions, *phos_E2F*, *init_RB*, and *inhib_E2F_RB_phos*, whereat *init_RB* is not connected to *phos_E2F* or *inhib_E2F_RB_phos*, which, in turn, are connected. There is a branching of the pathway, represented by the T-invariants

- *Inv1 = {init_CDK2,phosph_RB, phos_E2F, dephos_RB, init_compl,init_RB, inhib_E2F_RB_phos, init_E2F}*,
- *Inv2 = {init_CDK2, init_CDK4, phosp_RB, phos_E2F, t69.dephos_RB, init_compl, init_RB, inhib_E2F_RB_phos, init_E2F}*, and
- *Inv3= {init_CDK2, init_CDK6, phos_RB, phos_E2F, dephos_RB, init_compl, init_RB, inhib_E2F_RB_phos, init_E2F}*.

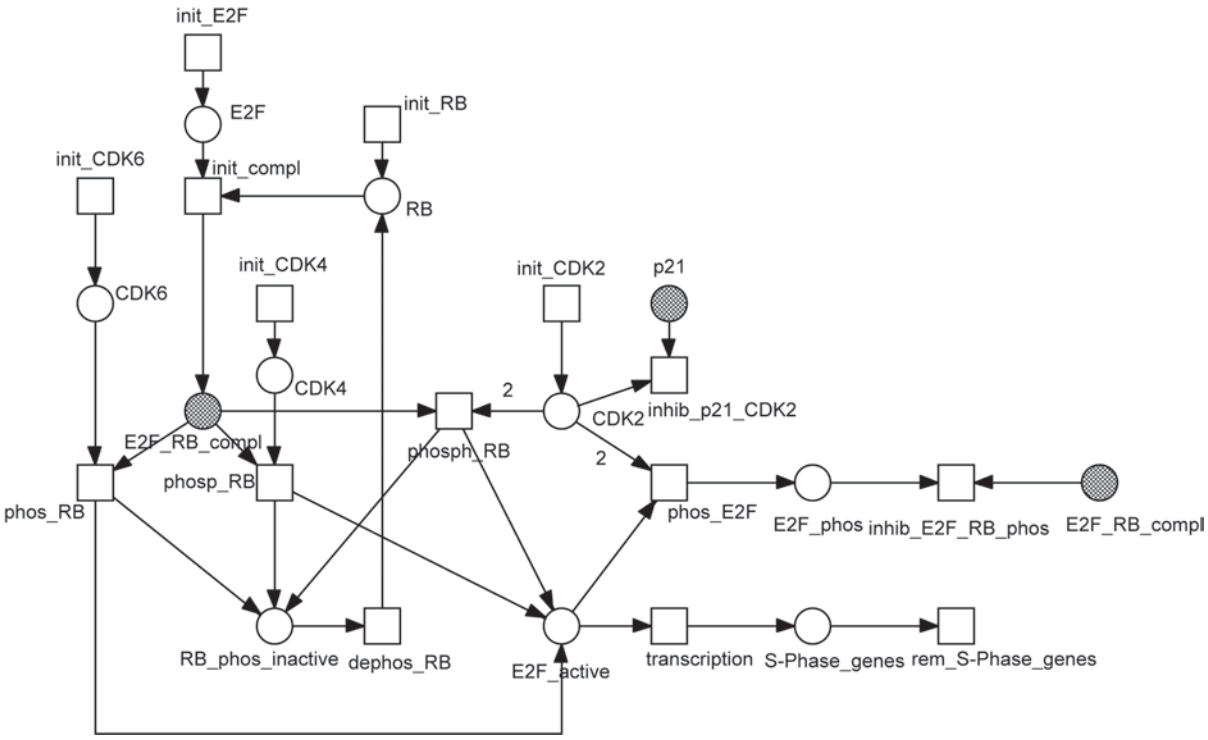
All invariants describe the process of inhibition of the E2F-RB complex via an initiation of RB. These T-invariants share the transitions in the MCT-set, which occur always together, and, exclusively, in these three T-invariants.

After generation of RB and E2F, forming the E2FRB-complex, the pathway can emerge in different ways due to the fact that three different kinases are able to phosphorylate the RB protein. *Inv1* describes the phosphorylation of RB and the retinoblastoma protein (*phosph_RB*) by CDK2. *Inv2* models phosphorylation by CDK4 (*phosp_RB*). *Inv3* includes phosphorylation of RB (phos_RB) by CDK6. Then, *E2F_active* phosphorylates E2F to form *E2F_phos,* and the successive pathway have all three transitions (*phos_E2F*, *init_RB*, *inhib_E2F_RB_phos*) in common. Thus, the three kinases, CDK2, CDK4, CDK6, responsible for phosphorylation of the same protein, are the reason for the disconnected MCT-set.

Table 1. The MCT-sets built by more than one transition. The asterisks mark the MCT-sets, M1 and M22, which describe disconnected sub-networks.

MCT-set	Composition	Biological interpretation
M1*	t0, t1, t6, t7, t8, t10, t30	Initiation of the Ca release out of the ER
M2	t2, t5	Ca-channel mediated Ca release depending on concentration gradient between ER and cytosol
M3	t3, t4	Ca release regulated by calreticulin
M4	t11, t12, t13, t14, t56, t57, t58	Activation of the DGC downstream pathway which activates JNK1
M5	t15, t17	Transcription of p21 by a transcription factor complex including NFATc
M6	t16, t20, t21, t22, t84	Transcription of MLC2, aActin, ANF by NFATc and other factors
M7	t18, t24, t40, t42, t85	Regulated transcription of UTRNA by NFATc and other factors
M8	t19, t25, t41, t43, t86	Regulated transcription of MYF5 by NFATc and other factors
M9	t26, t28	Down-regulation of RAP2B
M10	t27, t29	Up-regulation of RAP2B
M11	t31, t33, t34	Initiation of dystrophin followed by generation of DCG and simulation of DMD by DGC loss
M12	t35, t37	Down-regulation of CSNK1A1
M13	t36, t38	Up-regulation of CSNK1A1
M14	t39, t59	Inhibition of the p21 transcription by phosphorylated c-JUN
M15	t45, t47	Down-regulation of NFATc
M16	t49, t87	Up-regulation of calcineurin
M17	t50, t51	Down-regulation of calcineurin
M18	t52, t55	Up-regulation of JNK1
M19	t53, t54	Down-regulation of JNK1
M20	t63, t67	Initiation of CDK4 and phosphorylation of RB by CDK4
M21	t64, t66	Initiation of CDK6 and phosphorylation of RB by CDK6
M22*	t68, t71, t75	Initiation of RB, phosphorylation of E2F by CDK2, which inhibits RB phosphorylation (preservation of E2F-RB complex)
M23	t69, t70, t76	Initiation of E2F followed by initiation of the E2F-RB complex and dephosphorylation of RB
M24	t72, t73	Transcription of S-phase genes
M25	t77, t78, t79	Initiation, and phosphorylation of p53 by CSNK1A1 which regulates transcription of p21

Figure 9. Processes described by the disconnected MCT-set M22. The black shaded places indicate logical places.RB and E2F form the E2FRB-complex. Because three different kinases are able to phosphorylate the RB protein, the pathway can then emerge in different ways. Inv1 describes the phosphorylation of RB and the retinoblastoma protein (phosph_RB) by CDK2. Inv2 models phosphorylation by CDK4 (phosp_RB). Inv3 includes phosphorylation of RB (phos_RB) by CDK6. Then, E2F_active phosphorylates E2F to form E2F_phos, and the successive pathway have all three transitions (phos_E2F, init_RB, inhib_E2F_RB_phos) in common. Thus, the three kinases, CDK2, CDK4, CDK6, responsible for phosphorylation of the same protein, are the reason for the disconnected MCT-set.



To answer question 5, MCT-sets are interpreted as building blocks, thus, leading to a network reduction. The reduced network is depicted in Figure 10. It can be used, for example, to facilitate an exhaustive knockout analysis. Because the transitions in an MCT-set occur always together, it is sufficient to knockout only one arbitrary transition of an MCT-set.

For deepening the T-invariant's analysis, we compute T-clusters as proposed in (Grafahrend-Belau et al., 2006). We used the Tanimoto coefficient as distance measure between T-invariants and the UPGMA algorithm for clustering with a threshold of 65%. Thus, we yield 34 T-clusters, which are listed with their composition of MCT-sets and transitions in the file, clusters.txt, in the supplementary material (Koch, 2008a). Figure 11 depicts the cluster tree.

After exploration of T-invariants using MCT-sets and T-clusters, we can trust in our model that it is consistent and reflects the main biological behavior. The network reduction gives us an overview on functional building blocks (question 4 and 5).

To the last two questions we can respond by knockout analysis, which enables us to gain new insights into the system behavior.

Figure 10. The reduced PN model, whereat each MCT-set is drawn as a hierarchical node. Each MCT-set summarizes several transitions and the places in between.

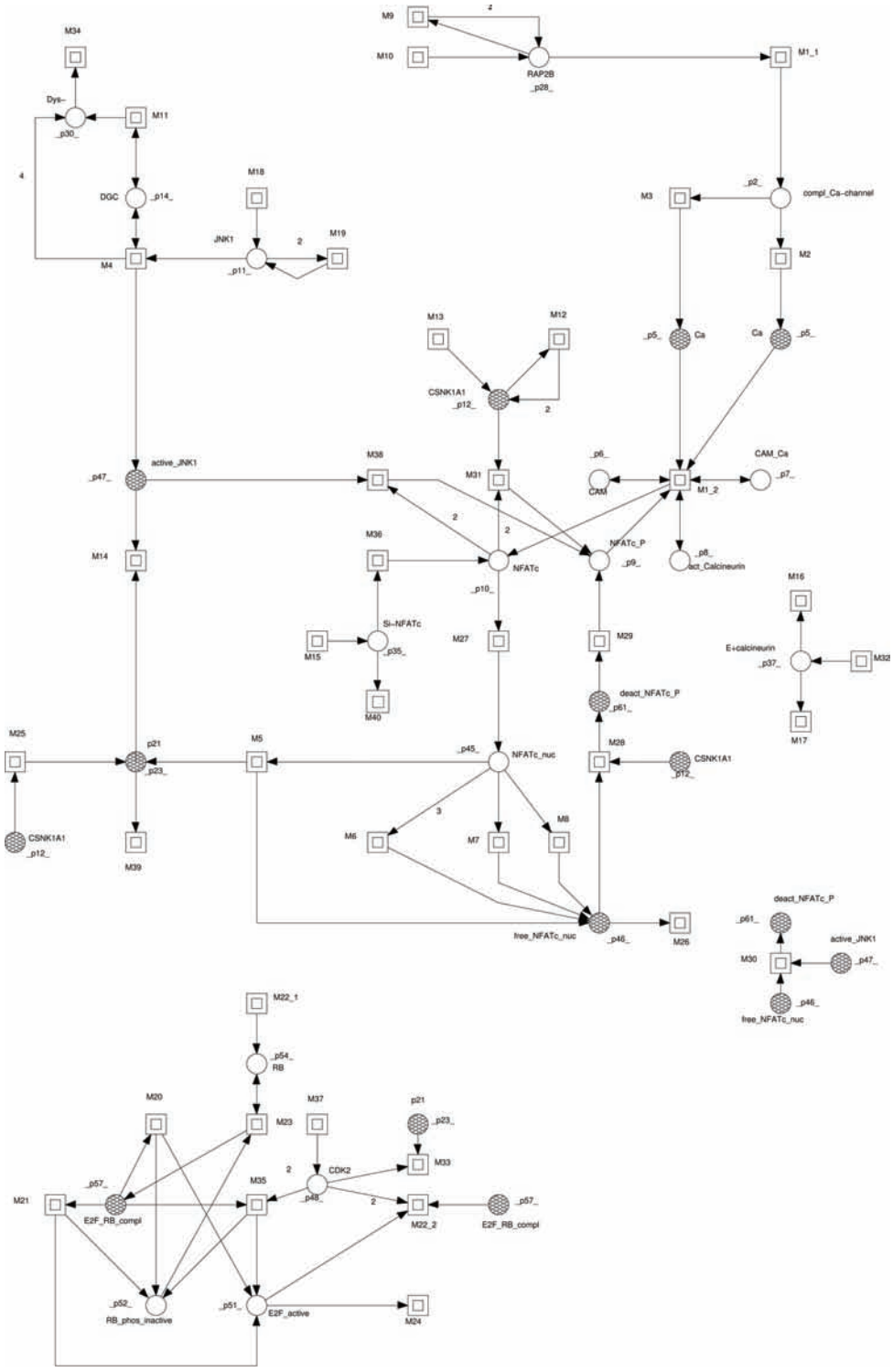


Figure 11. The clustering tree. The edges are labeled according to the distinguishing properties between T-clusters before the first branch and the common properties after the first branch. For example, T-cluster16 differs from T-Cluster17 in RAP2B regulation, but both clusters involve Ca release via calreticulin, and both, in turn, differ from T-clusters, 14 and 15, which both involve Ca release via the calcium channel.

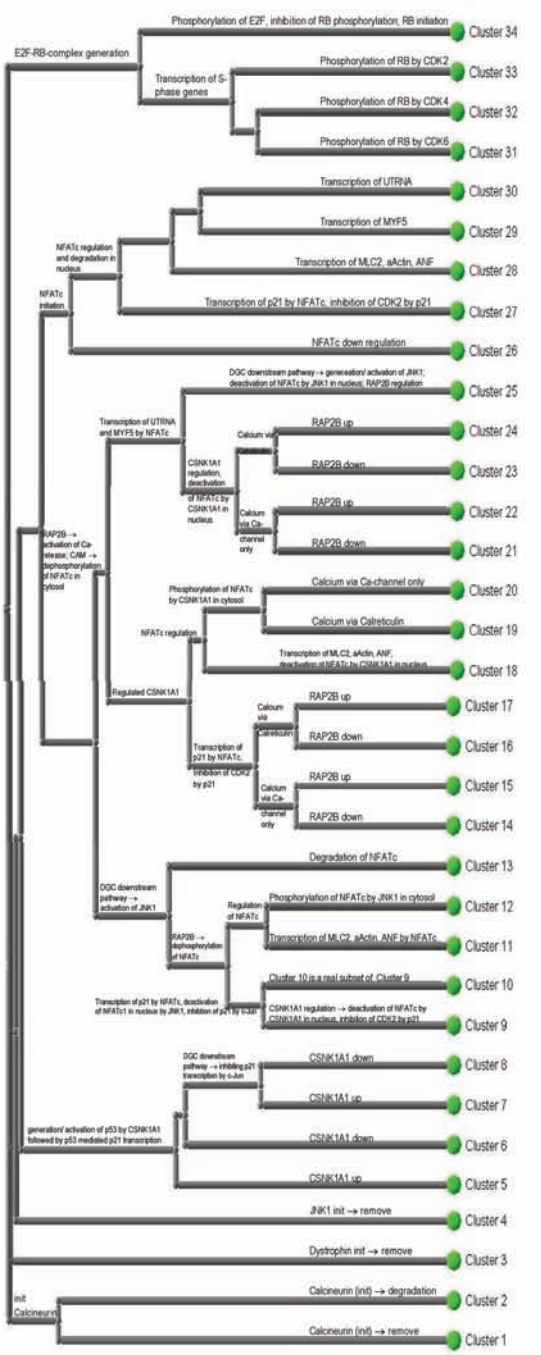


Table 2. The most important activities according to their knockout impact. The impact of a knockout is measured by the percentage of the number of T-invariants affected by it. Activities with a knockout impact below 20% are not listed.

MCT-set/transition	Activity	Knockout impact
M1	Dephosphorylation of NFATc	78%
t61	NFATc migrates into nucleus	73%
t83	Transportation of NFATc back to cytosol	67%
t81	Deactivation of NFATc by CSNK1A1	45%
M2	Ca release mediated by Ca-channel	39%
M3	Ca release regulated by calreticulin	39%
M9	Down-regulation of RAP2B	39%
M10	Up-regulation of RAP2B	39%
t32	Removal of dystrophin	37%
t52	Removal of silenced calcineurin	37%
t55	Initiation of enhancer of JNK1	37%
M4	Activation of the DGC pathway	36%
M5	Transcription of p21	36%
M12	Down-regulation of CSNK1A1	29%
M13	Up-regulation of CSNK1A1	29%
M15	Down-regulation of NFATc	29%
t46	Up-regulation of NFATc	28%
t82	Deactivation of NFATc by JNK1	22%

Let us first study the knockout impact defined by an MCT-set or a single transition. The impact of a knockout of a transition corresponds to the rate of reduction of the functional diversity of the system. It is determined by the percentage of the number of T-invariants affected by the knockout, assuming that the importance of a transition or of a set of transitions is related to the percentage of invariant destruction. In our PN model, the dephosphorylation of the transcription factor NFATc (*M1*), its migration into the nucleus (*Transfactor_in_nuc*), and the transport back to the cytosol (*Transfactor_in_cyt*), exhibit the largest impact, compare Table 2.

To study dependencies of the functional entities to each other, we use the Mauritius map representation. The entire Mauritius map is provided in the file, pn2_0803_fullb.pdf, supplementary material (Koch, 2008a).

To illustrate an example in more detail we want to restrict to the dominant part of the Mauritius map, where all edges of the tree with a relative knockout impact below 20% are dropped. Figure 12 depicts the corresponding Mauritius map.

The dominant part of the Mauritius map describes the interrelations of the molecular processes contributing to the dephosphorylation of the transcription factor NFATc. We can see that the right edge of the root exhibits the most important activity, i.e., the molecular processes of the MCT-set, *M1*, covering the transitions *act_PLCe.t30.M1*, *bind_PIP2.t0.M1*, *act_IP3.t1.M1*, *bind_CAM_Ca.t6.M1*, *gen_CAM.t7.M1*, *act_Calcineurin.t8.M1*, and *dephosph.t10.M1*, compare also Table 1. The central role of *M1* cor-

responds to the key role of the dephosphorylation, i.e., activation of NFATc (*dephosph.t10.M1*). Dephosphorylation of NFATc relies on the activation of PLCe by RAP2B (*act_PLCe.t30.M1*), the generation of calmodulin (*gen_CAM.t7.M1*), the binding of calcium to calmodulin (*bind_CAM_Ca.t6.M1*), and the activation of calcineurin by calmodulin (*act_Calcineurin.t8.M1*). All these activities are indispensable to the dephosphorylation of NFATc, whereas alternatives exist for other processes of calcium release from the endoplasmatic reticulum (ER).

The transitions *Transfactor_in_nuc.t61*, describing the migration of NFATc into the nucleus, and *Transfactor_in_cyt.t83*, describing the transport of NFATc back to the cytosol, represent the labels of the next right edge in the Mauritius map. They are the most important transitions for the functional part of the model, which depends on dephosphorylation of NFATc. In this context, both transitions together act as an MCT-set, having, therefore, identical knockout effects. The right edge denotes the deactivation of NFATc by CSNK1A1 (*deact_NFATc_CSNK1A1.t81*), which is a precondition for the transportation of NFATc back to the cytosol.

The left edge of *deact_NFATc_CSNK1A1.t81* is labeled by the molecular processes indispensable for the activation of JNK1 via the DGC downstream pathway, because the phosphorylation of NFATc is mediated either by CSNK1A1 or, via the DGC downstream pathway, by JNK1. Note that these processes only build an MCT-set in the context of the phosphorylation of NFATc, but may contribute in different ways to other activity groups of the net.

Let us now follow the child nodes of the activity of CSNK1A1 to phosphorylate, i.e., inactivate NFATc (*deact_NFATc_CSNK1A1.t81*). Deactivated NFATc cannot enter the nucleus, and would accumulate in the cytosol. The high level of phosphorylated NFATc in the cytosol can be compensated by an advanced activity of calcineurin to dephosphorylate NFATc. The activation of calcineurin is mediated by the calcium regulated calmodulin. The necessary calcium is stored in the ER.

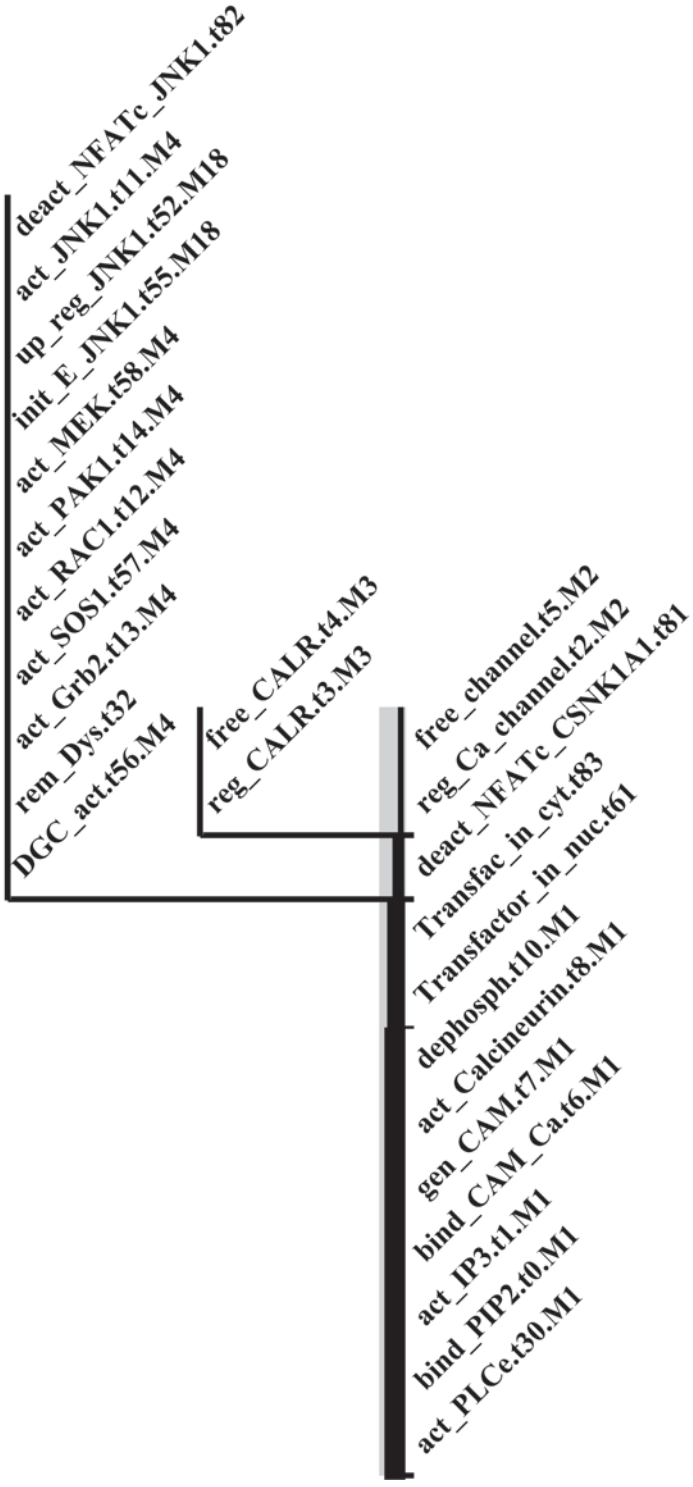
The molecular processes of the release of calcium from the ER into the cytosol, initiated by the concentration gradient between ER and cytosol (*M2*), or by regulated calreticulin (*M3*), are the labels of the following right and left edges, respectively. Both, *M2* and *M3*, represent alternative ways to provide free calcium for binding to calmodulin.

Through extensive PN analyses we found that the RAP2B–calcineurin pathway turned out to be one of the most relevant parts for the net behavior. A knockout in that arm would lead to a malfunction of 78% of the whole PN. The inhibition of calcineurin by cyclosporine A and FK506 causes profound bone loss in animal models suggesting a role of calcineurin in skeletal remodeling (Sun, 2005). Therefore, particularly for calcineurin, the phosphatase of our key component, NFATc, we conclude that an influence on calcineurin would have a big impact on NFATc mediated transcription, more precisely, on the reduction of p21 transcription. Latter can advance the increase of myoblast proliferation. This led in turn to experiments with chemical compounds such as cyclosporine A and okadaic acid, affecting calcineurin and phosphatases in general, respectively.

SUMMARY & CONCLUSION

This chapter aims at explaining the application of PN theory to model and analyze GRNs. To acquaint the reader with PNs, we give an introduction into basic definitions of P/T nets and their structural and dynamic properties using small biochemical examples. We explain which properties can be used to validate and investigate the system behavior.

Figure 12. The dominant part of the Mauritius map, i.e., only edges with a relative knockout impact above 20% are considered. The name of a transition is concatenated with its transition number and MCT-set.



The analysis is focused on exploring T-invariants, which decompose the net into a complete set of basic pathways. T-invariants describe always connected sub-networks with a special biological meaning.

For large and complex networks, which usually exhibit a large number of T-invariants, a further decomposition is possible by the concepts of MCT-sets and T-clusters. MCT-sets give disjoint sub-networks that can be disconnected, whereas T-invariants and T-clusters can overlap. Both concepts result in biologically meaningful functional modules and can be used for model validation.

For knockout analysis, we introduce the use of Mauritius map, which represent dependencies of transitions and MCT-sets in T-invariants, and give a useful visualization based on binary trees.

In contrast to other publications on modeling GRNs using PNs, we explore a much larger system of processes concerning gene regulation in DMD, which is mainly based on own experimental data. We explain the meaning of system's invariants and the results of their extended analysis in the biological context. Consequently, the PN and, in particular, its Mauritius map allow and, even more, simplify the search for candidate genes used in siRNA or vector-DNA experiments, additionally. Both techniques are specific for one gene, which makes it reasonable to have a tool for a selection of those candidate genes, which probably have the greatest impact on myoblasts of DMD patients. Using human primary myoblasts, an increase as well as a knockout of several proteins of the net is thought for validation of the network itself and its analysis results followed by examinations on mRNA and protein level to complete modeling of expression regulation.

A first great, often underestimated, advantage in using PNs is the intuitive graphical visualization allowing for hierarchical modeling. This point is in particular useful in the communication between computer scientists or mathematicians and biologist or medical scientists. The second great advantage is the variety of different analysis techniques to explore the net behavior based on structural as well as on dynamic properties.

The main advantage of PN modeling is the ability to model on the one hand at different description levels, and on the other hand to combine these description levels into one model. Discrete PNs can yield information about system's consistency and system behavior without any knowledge of kinetic data. In particular for GRNs, where often kinetic data are missed, discrete modeling is the only way to get some insights into system behavior.

Quantitative modeling using PNs, usually based on ODEs, provide at the moment no significant advantage compared to software packages, well-established in systems biology. These packages are purpose-built for biological applications such that they often provide interfaces to SBML (Hucka, 2003), links to special pathway, enzyme, sequence and structure data bases, and also to literature data bases.

Hybrid PNs provide a powerful possibility to combine modeling at discrete and continuous level, especially for those cases with incomplete kinetic data. Software tools as GON (Nagasaki, 2003), (Doi, 2003) and its commercial version Cell Illustrator (Poland, 2008) have been applied to simulate many different biochemical systems, providing a special visualization mode for analyzing and simulating GRNs.

A disadvantage and, thus, also a challenge is the lack of PN tools designed particularly for biological applications. In biology, there are many well-established terms and representations that can be translated to PNs. Because of the many exceptions in biology, a frequently changing knowledge, many incomplete data and a huge amount of data, biochemical models of interest are very complex and large. There are still limitations in exploring large biochemical systems, in particular for gene expression data of some thousands of genes.

ACKNOWLEDGMENT

I'm deeply grateful to Stefanie Grunwald and Astrid Speer for their work on the experimental part of the case study and for many interesting discussions. I thank Jörg Ackermann, who is mainly responsible for the Mauritius maps. Also, I would like to acknowledge Ina Weiss for her help and valuable suggestions. Last, but not least, I want to thank Martin Vingron for his support of my work.

REFERENCES

Alla, H., & David, R. (1998). Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, *8*(1), 159–188. doi:10.1142/S0218126698000079

Backhaus, K., Erichson, B., Plinke, W., & Weiber, R. (2003). *Multivariate analysis methods. An application-oriented introduction (in German)* (10th ed.). Berlin: Springer-Verlag.

Baumgarten, B. (1996). *Petri-Netze: Grundlagen und Anwendungen (in German)* (2nd ed.). Heidelberg, Berlin, Oxford: Spektrum Akademischer Verlag GmbH.

Bause, F., & Kritzinger, P. (1996). *Stochastic Petri nets: An introduction to the theory*. Vieweg-Verlag.

Billington, J., Diaz, M., & Rozenberg, G. (Eds.). (1999). *Application of Petri nets to communication networks* (Vol. 1605). Berlin: Springer-Verlag.

Chaouiya, C. (2007). Petri net modelling of biological networks. *Briefings in Bioinformatics*, *8*(4), 210–219. doi:10.1093/bib/bbm029

Chen, M., & Hofestädt, R. (2003). Quantitative Petri net model of gene regulated metabolic networks in the cell. In *Silico Biology, 3*.

David, R., & Alla, H. (2005). *Discrete, continous, and hybrid Petri nets*. Berlin: Springer-Verlag.

Doi, A., Masao Nagasaki, M., Matsuno, H., & Satoru Miyano, S. (2003). Genomic object net II: How to model biopathways with hybrid functional Petri net with extension. *Applied Bioinformatics, 2*(3), 185–188.

Edwards, J. S., P. B. O. (2000). The Escherichia coli MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(10), 5528–5533. doi:10.1073/pnas.97.10.5528

Endesfelder, S., Bucher, S., Kliche, A., Reszka, R., & Speer, A. (2003). Transfection of normal primary human skeletal myoblasts with p21 and p57 antisense oligonucleotides to improve their proliferation: A first step towards an alternative molecular therapy approach of Duchenne muscular dystrophy. *Journal of Molecular Medicine*, *81*, 355–362.

Endesfelder, S., Kliche, A., Lochmüller, H., von Moers, A., & Speer, A. (2005). Antisense oligonucleotides and short interfering RNAs silencing the cyclin-dependent kinase inhibitor p21 improve proliferation of Duchenne muscular dystrophy patients' primary skeletal myoblasts. *Journal of Molecular Medicine*, *83*, 64–71. doi:10.1007/s00109-004-0607-3

Esparza, J., & Nielsen, M. (1994). Decidabilty issues for Petri nets. *Journal of Information Processing and Cybernetics, EIK*, *30*, 143–160.

Genrich, H., Küffner, R., & Voss, K. (2001). Executable Petri net models for the analysis of metabolic pathways. *International Journal on Software Tools for Technoloy Transfer*, *4*, 1–11.

Goss, P. J. E., & Peccoud, J. (1998). Quantitative modeling of stochastic systems in moldecular biology by using stochstic Petri nets. *Proceedings of the National Academy of Sciences of the United States of America*, *95*, 6750–6755. doi:10.1073/pnas.95.12.6750

Goss, P. J. E., & Peccoud, J. (1999). *Analysis of the stabilizing effect of Rom on the genetic network controlling ColE1 plasmid replication.* Paper presented at the Pacific Symposium on Biocomputing, Hawaii.

Grafahrend-Belau, E. (2006). *Classification of T-Invariants in biochemical Petri nets based on different cluster analysis techniques (in German).* Technical University of Applied Sciences Berlin, Berlin.

Grafahrend-Belau, E., Schreiber, F., Heiner, M., Sackmann, A., Junker, B. H., & Grunwald, S. (2008). Modularization of biochemical networks based on classification of Petri net t-invariants. *Journal*, *9*, 90. doi:.doi:10.1186/1471-2105-9-90

Grunwald, S., Speer, A., Ackermann, J., & Koch, I. (2008). Petri net modelling of gene regulation of the Duchenne muscular dystrophy. *Bio Systems*, *92*, 189–205. doi:10.1016/j.biosystems.2008.02.005

Haas, P. J. (2002). *Stochastic Petri nets, modelling, stability, simulation*. Berlin: Springer-Verlag.

Hack, M. H. T. (1976). *Decidability questions for Petri nets.* M.I.T.

Hardy, S., R. P. N. (2004). *J. Bioinform. Comput. Biol.*,*2*(4), 595-613doi:10.1142/S0219720004000764

Hardy, S., & Robillard, P. N. (2004). Modeling and simulation of molecular biology systems using Petri nets: Modeling goals of various approaches. *Journal of Bioinformatics and Computational Biology*, *2*(4), 619–637. doi:10.1142/S0219720004000764

Heiner, M., & Koch, I. (2004). *Petri net based model validation in systems biology.* Paper presented at the Proceedings of the 25th International Conference on Applications and Theory of Petri Nets (ICATPN 04), Bologna.

Hofestädt, R. (1994). A Petri net application of metabolic processes. *Journal of Systems Analysis . Modelling and Simulation (Anaheim)*, *16*, 113–122.

Hofestädt, R., & Thelen, S. (1998). Quantitative modeling of biochemical networks. *In Silico Biology*, *1*(1), 39–53.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., & Kitano, H. (2003). The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics (Oxford, England), 19*(4), 524–531. doi:10.1093/bioinformatics/btg015

Kielbassa, J., Bortfeldt, R., Schuster, S., & Koch, I. (2008). Modeling of the U1 snRNP assembly pathway in alternative splicing in human cells using Petri nets. *Computational Biology and Chemistry*.

Koch, I. (2008a). *Supplementary material to modeling Duchenne muscular dystrophy using Petri nets*. Retrieved from http://www.molgen.mpg.de/~koch_i/

Koch, I. (2008b). *Supplementary material*. Retrieved from http://www.molgen.mpg.de/~koch_i/

Koch, I., & Heiner, M. (2008). Petri nets in biological network analysis. In B. Junker & F. Schreiber (Eds.), *Analysis of biological networks* (pp. 139–179). Wiley & Sons.

Koch, I., Junker, B., & Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics (Oxford, England), 21*(7), 1219–1226. doi:10.1093/bioinformatics/bti145

Larhlimi, A., & Bockmayr, A. (2006). *A new approach to flux coupling analysis of metabolic networks.* Paper presented at the Computational Life Sciences II, CompLife'06, Cambridge, UK.

Lee, D.-Y., Zimmer, R., Lee, D.-Y., Hanisch, D., & Sunwon, P. (2004). Knowledge representation model for systems-level analysis of signal transduction networks. *Genome Informatics, 15*(2), 234–243.

Marwan, W., Sujathab, A., & Starostzik, C. (2005). Reconstructing the regulatory network controlling commitment and sporulation in Physarum polycephalum based on hierarchical Petri net modeling and simulation. *Journal of Theoretical Biology, 236*, 349–365. doi:10.1016/j.jtbi.2005.03.018

Matsuno, H., Doi, A., Nagasaki, M., & Miyano, S. (2000). *Hybrid Petri net representation of gene regulatory network.* Paper presented at the Pacific Symposium on Biocomputing, Hawaii.

Matsuno, H., Li, C., & Miyano, S. (2006). Petri net based descriptions for systematic understanding of biological pathways. *IEICE Transactions on Fundamentals of Electronics, Communication, and Computer Sciences . E (Norwalk, Conn.), 89-A*(11), 3166–3174.

Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., & Miyano, S. (2003). Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biology, 3*(3), 389–404.

Murata, T. (1989). *Petri Nets: Properties, analysis, and applications.* Paper presented at the IEEE-International Conference on Consumer Electronics. Digest of Technical Papers.

Nagasaki, M., Atsushi Doi, A., Matsuno, H., & Miyano, S. (2003). Genomic object net I: A platform for modeling and simulating biopathways. *Applied Bioinformatics, 2*(3), 181–184.

Nutsch, T., Oesterhelt, D., Gilles, E. D., & Marwan, W. (2005). A quantitative model of the switch cycle of an archaeal flagellar motor and its sensory control. *Biophysical Journal, 89*, 2307–2323. doi:10.1529/biophysj.104.057570

Oliveira, J. S., Bailey, C. G., Jones-Oliveira, J. B., Dixon, D. A., Gull, D. W., & Chandel, M. L. (2003). A computational model for the identification of biochemical pathways in the Krebs cycle. *Journal of Computational Biology*, *10*(1), 57–82. doi:10.1089/106652703763255679

Parikh, R. J. (1966). On context-free languages. *Journal of the association for computing machinery, 13*, 570–581.

Peleg, M., Rubin, D., & Altman, R. B. (2005). Using Petri net tools to study properties and dynamics of biological systems. *Journal of the American Medical Informatics Association*, *12*(2), 181–199. doi:10.1197/jamia.M1637

Peterson, J. L. (1981). *Petri net theory and the modeling of systems*. Englewood Cliffs, NJ: Prentice Hall.

Petri, C. A. (1962). *Communication with automata (in German).* TU Darmstadt, Darmstadt.

Poland, F. (2008). Cell illustrator. Retrieved from http://www.fqs.pl/life_science/cell_illustrator

Popova-Zeugmann, L., Heiner, M., & Koch, I. (2005). Modelling and analysis of biochemical networks with time Petri nets. *Fundamenta Informaticae, 67,* 149–163.

Proth, J.-M., & Xie, X. (1997). *Petri nets: A tool for design and management of manufacturing systems*. John Wiley & Sons, Inc.

Reddy, V. N. (1994). *Modeling biological pathways: A discrete event systems approach.* University of Maryland.

Reddy, V. N., Mavrovouniotis, M. L., & Liebman, M. N. (1993). *Petri net representations in metabolic pathways.* Paper presented at the ISMB International Conference on Intelligent Systems in Molecular Biology Bethesda, MD.

Reddy, V. N., Mavrovouniotis, M. L., & Liebman, M. N. (1996). Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, *26*(1), 9–24. doi:10.1016/0010-4825(95)00042-9

Reisig, W. (1985). *Petri nets: An introduction* (2nd ed., Vol. 4). Berlin Springer-Verlag.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. doi:10.1016/0377-0427(87)90125-7

Sackmann, A. (2005). *Modeling and simulation of signal transduktion pathways in Saccharomyces cerevisiae based on Petri net theory (in German).* Technical University of Applied Sciencesm, Ernst-Moritz-Arndt-University Greifswald, Berlin, Greifswald.

Sackmann, A., Formanowicz, D., Formanowicz, P., Koch, I., & Blazewicz, J. (2007). An analysis of the Petri net based model of the human body iron homeostasis process. *Computational Biology and Chemistry*, *31*, 1–10. doi:10.1016/j.compbiolchem.2006.09.005

Sackmann, A., Heiner, M., & Koch, I. (2006). Application of Petri net based analysis techniques to signal transduction pathways. *Journal*, *7*, 482. doi:.doi:10.1186/1471-2105-7-482

Saito, A., Nagasaki, M., Doi, A., Ueno, K., & Miyano, M. (2006). Cell fate simulation model of gustatory neurons with microRNAs double-negative feedback loop by hybrid functional Petri net with extension. *Genome Informatics*, *17*(1), 100–111.

Saitou, N., & Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, *4*, 406–425.

Schilling, C. H., C. M. W., Famili, I., Church, G. M., Edwards, J. S., & Palsson, B.O. (2002). Genome-scale metabolic model of Helicobacter pylori 26695. *Journal of Bacteriology*, *184*, 4582–4593. doi:10.1128/JB.184.16.4582-4593.2002

Schilling, C. H., Letscher, D., & Palsson, B. O. (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway oriented perspective. *Journal of Theoretical Biology*, *203*, 229–248. doi:10.1006/jtbi.2000.1073

Schuster, S., Hilgetag, C., & Schuster, R. (1993). *Determining elementary modes of functioning in biochemical reaction networks at steady-state.* Paper presented at the Second Gauss Symposium.

Simao, E., Remy, E., Thieffry, D., & Chaouiya, C. (2005). Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in E.Coli. *Bioinformatics (Oxford, England)*, *21*(Suppl. 2), ii190–ii196. doi:10.1093/bioinformatics/bti1130

Starke, P. H. (1990). *Analysis of Petri net models (in German)*. Stuttgart: B. G. Teubner.

Steggles, L. J., Banks, R., & Wipat, A. (2006). *Modelling and analysing genetic networks: From Boolean networks to Petri nets.* Paper presented at the Computational Methods in Systems Biology.

Steggles, L. J., Banks, R., & Wipat, A. (2007). Qualitatively modelling and analysing genetic regulatory networks: A Petri net approach. *Bioinformatics (Oxford, England)*, *23*(3), 336–343. doi:10.1093/bioinformatics/btl596

Steinhausen, D., & Langer, K. (1977). *Cluster analysis: An introduction to methods for automatic classification (in German)*. Berlin: de Gruyter.

Studier, J. A., & Keppler, K. J. (1988). A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution*, *5*, 729–731.

Sun, L., Blair, H. C., Peng, Y., Zaidi, N., Adebanjo, O. A., & Wu, X. B. (2005). Calcineurin regulates bone formation by the osteoblast. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 17130–17135. doi:10.1073/pnas.0508480102

Takai-Igarashi, T. (2005). Ontology based standardization of Petri net modeling for signalling pathways. *Journal*, *5*, 0047

TGI-group. (2008). *Petri Nets World*. Retrieved from http://www.informatik.uni-hamburg.de./TGI/PetriNets

Thomas, R., Thieffry, D., & Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks–I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, *57*, 247–276.

Valmari, A. (1998). *In lectures in Petri nets I: The state explosion problem* (Vol. 1491). Berlin: Springer.

van der Aalst, W. M. P., Desel, J., & Oberweis, A. (1999). *Business process management: Models, techniques, and empirical studies* (Vol. 1806). Berlin: Springer-Verlag

Voss, K., Heiner, M., & Koch, I. (2003). Steady state analysis of metabolic pathways using Petri nets. *In Silico Biology*, *3*(3), 367–387.

KEY TERMS AND DEFINITIONS

Petri Net Theory Facilitates Modeling: Analysis, and simulation of biochemical networks at different abstraction levels

The benefit of applying Petri nets to biology is the qualitative discrete analysis allowing for prediction of the dynamic net behavior.: In particular for GRNs: Petri net provide useful analysis techniques, as invariant analysis, MCT-sets, T-clusters, and Mauritius maps

Mauritius maps support *in silico* **knockout analyses of biochemical systems.: There are many other interesting powerful applications of Petri nets to biochemical networks:** and in particular to GRNs, as Boolean based Petri nets, hybrid Petri nets, continuous Petri nets, and stochastic Petri nets

Compilation of References

Ackers, G. K., Johnson, A. D., & Shea, M. A. (1982). Quantitative model for gene regulation by λ phage repressor. *Proceedings of the National Academy of Sciences of the United States of America*, *79*, 1129–1133. doi:10.1073/pnas.79.4.1129

Adami, C., Ofria, C., & Collier, T. C. (2000). Evolution of biological complexity. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(9), 4463–4468. doi:10.1073/pnas.97.9.4463

Adamic, L. A., & Huberman, B. A. (1999). Growth dynamics of the World Wide Web. *Nature*, *401*, 131. doi:10.1038/43604

Aerts, S., van Helden, J., Sand, O., & Hassan, B. A. (2007). Fine-tuning enhancer models to predict transcriptional targets across multiple genomes. *PLoS ONE*, *2*(11), e1115. doi:10.1371/journal.pone.0001115

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *AC-19*, 716–723. doi:10.1109/TAC.1974.1100705

Akutsu, T., Kuhara, S., Maruyama, O., & Miyano, S. (2003). Identification of genetic networks by strategic gene disruptions and gene overexpressions under a Boolean model. *Theoretical Computer Science*, *298*, 235–251. doi:10.1016/S0304-3975(02)00425-5

Akutsu, T., Miyano, S., & Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing* (pp. 17-28).

Akutsu, T., Miyano, S., & Kuhara, S. (2000). Algorithms for inferring qualitative models of biological networks. *Pacific Symposium on Biocomputing, 5*, 290-301.

Akutsu, T., Miyano, S., & Kuhara, S. (2000). Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics (Oxford, England)*, *16*, 727–734. doi:10.1093/bioinformatics/16.8.727

Ala, U., Piro, R. M., Grassi, E., Damasco, C., Silengo, L., & Oti, M. (2008). Prediction of human disease genes by human-mouse conserved coexpression analysis. *PLoS Computational Biology*, *4*(3), e1000043. doi:10.1371/journal.pcbi.1000043

Alaoui-Ismaili, M. H., Lomedico, P. T., & Jindal, S. (2002). Chemical genomics: discovery of disease genes and drugs. *Drug Discovery Today*, *7*(5), 292–294. doi:10.1016/S1359-6446(02)02185-2

Albert, J., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, *88*, 669–679. doi:10.2307/2290350

Albert, R. (2004). Boolean modeling of genetic regulatory networks. In E. Ben-Naim, H. Frauenfelder & Z. Toroczkai (Eds.), *Complex networks*. (LNP, pp. 459-479). Berlin: Springer.

Albert, R. (2005). Scale-free networks in cell biology. *Journal of Cell Science*, *118*, 4947–4957. doi:10.1242/jcs.02714

Albert, R., & Barabási, A. L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*, 47–91. doi:10.1103/RevModPhys.74.47

Albert, R., & Barabasi, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*, 47–97. doi:10.1103/RevModPhys.74.47

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., & Walter, P. (2002). *Molecular biology of the cell, fourth edition*. Garland Science, Taylor & Francis Group.

Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D. Nonlinear Phenomena*, *185*, 45–66. doi:10.1016/S0167-2789(03)00174-X

Aldana, M., Balleza, E., Kauffman, S. A., & Resendiz, O. (2006). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology, 245,* 433–448. doi:10.1016/j.jtbi.2006.10.027

Aldana, M., Coppersmith, S., & Kadanoff, L. P. (2003). Boolean dynamics with random couplings. In E. Kaplan, J. E. Marsden & K. R. Sreenivasan (Eds.), *Perspectives and problems in nonlinear science.* Springer Applied Mathematical Sciences Series (pp. 23-89). Berlin: Springer.

Alla, H., & David, R. (1998). Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers, 8*(1), 159–188. doi:10.1142/S0218126698000079

Almasri, E., Larsen, P., Chen, G., & Dai, Y. (2008). Incorporating literature knowledge in Bayesian network for inferring gene networks with gene expression data. *4th International Symposium on Bioinformatics Research and Applications* (pp. 184-195). Springer-Verlag.

Alon, U. (2007). *An introduction to systems biology. Design principles of biological circuits.* London: Chapman & Hall.

Alon, U. (2007). Network motifs: Theory and experimental approaches. *Nature Genetics, 8,* 450–461. doi:10.1038/nrg2102

Amaral, L. A. N., Diaz-Guilera, A., Moreira, A. A., Goldberger, A. L., & Lipsitz, L. A. (2004). Emergence of complex dynamics in a simple model of signaling networks. *Proceedings of the National Academy of Sciences of the United States of America, 101,* 15551. doi:10.1073/pnas.0404843101

Amaral, L., Scala, A., Barthélémy, M., & Stanley, H. (2000). Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America, 97*(21), 11149–11152. doi:10.1073/pnas.200327197

Ambros, V. (2004). The function of animal microRNAs. *Nature, 431,* 350–355. doi:10.1038/nature02871

an der Heiden, U. (1979). Delays in physiological systems. *Journal of Mathematical Biology, 8,* 345–364.

Ando, S., & Iba, H. (2001). Inference of gene regulatory model by genetic algorithms. In *Proceedings of the 2001 Congress on Evolutionary Computation, 1,* 712-719. IEEE-Press.

Anholt, R. R., Dilda, C. L., Chang, S., Fanara, J. J., Kulkarni, N. H., & Ganguly, I. (2003). The genetic architecture of odor-guided behavior in Drosophila: Epistasis and the transcriptome. *Nature Genetics, 35*(2), 180–184. doi:10.1038/ng1240

Arenas, A., Diaz-Guilera, A., & Perez-Vicente, C. J. (2006). Synchronization reveals topological scales in complex networks. *Physical Review Letters, 96,* 114102. doi:10.1103/PhysRevLett.96.114102

Arkin, A., & Fletcher, D. (2006). Fast, cheap, and somewhat in control. *Genome Biology, 7*(8), 114. doi:10.1186/gb-2006-7-8-114

Arkin, A., Ross, J., & McAdams, H. H. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli cells. Genetics, 149,* 1633–1648.

Arkin, A., Shen, P., & Ross, J. (1997). A test case of correlation metric construction of a reaction pathway from measurements. *Science, 277,* 1275–1279. doi:10.1126/science.277.5330.1275

Arnone, M. I., & Davidson, E. H. (1997). The hardwiring of development: Organization and function of genomic regulatory systems. *Development, 124,* 1851–1864.

Ascher, U. M., & Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential-algebraic equations.* SIAM.

Ashburner, M., Ball, C. A., Blake, J. A., & Botstein, D. (2000). Gene ontology: Tool for the unification of biology. The gene ontology consortium. *Nature Genetics, 25,* 25–29. doi:10.1038/75556

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., et al. (2000). Gene ontology: Tool for the unification of biology. *Naure Genetics,* 25, 25–29. Annotations retrieved in December 2007, from http://www.yeastgenome.org/

Atay, F. M., & Biyikoglu, T. (2005). Graph operations and synchronization of complex networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 72,* 016217. doi:10.1103/PhysRevE.72.016217

Atkinson, M., Savageau, M., Myers, J., & Ninfa, A. (2003). Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in Escherichia coli. *Cell, 113,* 597–607. doi:10.1016/S0092-8674(03)00346-5

Aurell, E., & Sneppen, K. (2002). Epigenetics as a first exit problem. *Physical Review Letters, 88*(048101).

Babu, M. M., Luscombe, N. M., Aravind, L., Gerstein, M., & Teichmann, S. A. (2004). Structure and evolution of transcriptional regulatory networks. *Current Opinion in Structural Biology, 14*, 283–291. doi:10.1016/j.sbi.2004.05.004

Backhaus, K., Erichson, B., Plinke, W., & Weiber, R. (2003). *Multivariate analysis methods. An application-oriented introduction (in German)* (10th ed.). Berlin: Springer-Verlag.

Baeza-Yates, R., & Ribeiro, B. A. N. (1999). *Modern information retrieval*. New York: Addison-Wesley.

Bajec, I. L., Zimic, N., & Mraz, M. (2005). Simulating flocks on the wing: The fuzzy approach. *Journal of Theoretical Biology, 233*, 199–220. doi:10.1016/j.jtbi.2004.10.003

Baker, D., & Sali, A. (2001). Protein structure prediction and structural genomics. *Science, 294*(5540), 93–96. doi:10.1126/science.1065659

Bakk, A., Metzler, R., & Sneppen, K. (2004). Sensitivity of OR in phage λ. *Biophysical Journal, 86*, 58–66. doi:10.1016/S0006-3495(04)74083-7

Balaji, S., Babu, M. M., Iyer, L. M., Luscombe, N. M., & Aravind, L. (2006). Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast. *Journal of Molecular Biology, 360*(1), 213–227. doi:10.1016/j.jmb.2006.04.029

Balázsi, G., Barabási, A. L., & Oltvai, Z. N. (2005). Topological units of environmental signal processing in the transcriptional regulatory network of Escherichia coli. *Proceedings of the National Academy of Sciences of the United States of America, 102*, 7841. doi:10.1073/pnas.0500365102

Balsalobre, A., Damiola, F., & Schibler, U. (1998). A serum shock induces circadian gene expression in mammalian tissue culture cells. *Cell, 93*, 929–937. doi:10.1016/S0092-8674(00)81199-X

Banerjee, A., & Jost, J. (2007). Spectral plots and the representation and interpretation of biological data. *Theory in Biosciences, 126*, 15–21. doi:10.1007/s12064-007-0005-9

Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. In R. Riolo & B. Worzel (Eds.), *Genetic programming series, 6*, 43-62. Springer Verlag.

Barabási, A. L., & Albert, R. (1999). Emergence of scaling in random networks. *Science, 286*, 509–512. doi:10.1126/science.286.5439.509

Barabási, A. L., & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Reviews. Genetics, 5*(2), 101–113. doi:10.1038/nrg1272

Bar-Joseph, Z., Gerber, G. K., Lee, T. I., Rinaldi, N. J., Yoo, J. Y., & Robert, F. (2003). Computational discovery of gene modules and regulatory networks. *Nature Biotechnology, 21*(11), 1337–1342. doi:10.1038/nbt890

Barker, N., Myers, C., & Kuwahara, H. (2006). Learning genetic regulatory network connectivity from time series data. In *The 19th International Conference on Industrial, Engineering, and Other Applications of Applied Intelligent Systems.*

Barrio, M., Burrage, K., & Leier, A. (2006). Oscillatory regulation of Hes1: Discrete stochastic delay modelling and simulation. *PLoS Comp. Bio., 2*(9), e117.

Basso, K., Margolin, A., Stolovitzky, G., Klein, U., Dalla-Favera, R., & Califano, A. (2005). Reverse engineering of regulatory networks in human B cells. *Nature Genetics, 37*, 382–390. doi:10.1038/ng1532

Batagelj, V., & Brandes, U. (2005). Efficient generation of large random networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 71*, 036113. doi:10.1103/PhysRevE.71.036113

Batagelj, V., & Mrvar, A. (2003). Pajek-analysis and visualization of large networks. In M. Jünger & P. Mutzel (Eds.), Graph drawing software (pp. 77-103). Springer.

Battle, A., Segal, E., & Koller, D. (2005). Probabilistic discovery of overlapping cellular processes and their regulation. *Journal of Computational Biology, 12*(7), 909–927. doi:10.1089/cmb.2005.12.909

Baumgarten, B. (1996). *Petri-Netze: Grundlagen und Anwendungen (in German)* (2nd ed.). Heidelberg, Berlin, Oxford: Spektrum Akademischer Verlag GmbH.

Bause, F., & Kritzinger, P. (1996). *Stochastic Petri nets: An introduction to the theory.* Vieweg-Verlag.

Bay, S., Chrisman, L., Pohorille, A., & Shrager, J. (2004). Temporal aggregation bias and inference of causal regulatory networks. *Journal of Computational Biology, 11*, 971–985. doi:10.1089/cmb.2004.11.971

Beal, M. J., Falciani, F., Ghahramani, Z., Rangel, C., & Wild, D. L. (2005). A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics (Oxford, England), 21*(3), 349–356. doi:10.1093/bioinformatics/bti014

Becskei, A., & Serrano, L. (2000). Regulation of noise in the expression of a single gene. *Nature*, *405*, 590–593. doi:10.1038/35014651

Bedrick, E. J., Christensen, R., & Johnson, W. (1996). A new perspective on priors for generalized linear models . *Journal of the American Statistical Association*, *91*, 1450–1461. doi:10.2307/2291571

Beer, M. A., & Tavazoie, S. (2004). Predicting gene expression from sequence. *Cell*, 16, *117*(2), 185-98.

Beinlich, I. A., Suermondt, H., Chavez, R., & Cooper, G. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. *Second European Conference in Artificial Intelligence in Medicine* (pp. 247-256).

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B. Methodological*, *57*, 289–300.

Bennett, W., Hussain, S., Vahakangas, K., Khan, M., Shields, P., & Harris, C. (1999). Molecular epidemiology of human cancer risk: Gene-environment interactions and p53 mutation spectrum in human lung cancer. *The Journal of Pathology*, *187*(1), 8–18. doi:10.1002/(SICI)1096-9896(199901)187:1<8::AID-PATH232>3.0.CO;2-Y

Benson, M. L., Smith, R. D., Khazanov, N. A., Dimcheff, B., Beaver, J., & Dresslar, P. (2007). Binding MOAD, a high-quality protein ligand database. *Nucleic Acids Research*, *36*(Database issue), D674–D678. doi:10.1093/nar/gkm911

Bentley, P., & Kumar, S. (1999). Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. *Proceedings of the Genetic and Evolutionary Computations Conference 1999,* Orlando, Florida (pp. 35-43).

Bernard, A., & Hartemink, A. J. (2005). Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. *Pac. Symp. Biocomp.* (PSB05) (pp. 459-470). NJ: World Scientific.

Bernard, S., Čajavec, B., & Pujo-Menjouet, L. (2006). Modeling transcriptional feedback loops: The role of Gro/LTE1 in hes1 oscillations. *Phil. Transact. A Math. Phys. Engineering and Science*, *364*, 1155–1170.

Bernardo, J. M., & Smith, A. F. M. (1994). *Bayesian theory. Wiley series in probability and mathematical statistics.* Chichester: John Wiley and Sons.

Berry, R. S., Rice, S. A., & Ross, J. (2000). *Physical chemistry* (2nd edition). New York: Oxford University Press.

Beyene, J., Tritchler, D., Bull, S. B., Cartier, K. C., Jonasdottir, G., & Kraja, A. T. (2007). Multivariate analysis of complex gene expression and clinical phenotypes with genetic marker data. *Genetic Epidemiology*, *31*(Suppl 1), S103–S109. doi:10.1002/gepi.20286

Beyer, A., Bandyopadhyay, S., & Ideker, T. (2007). Integrating physical and genetic maps: From genomes to interaction networks. *Nature Reviews. Genetics*, *8*, 699–710. doi:10.1038/nrg2144

Beyer, H., & Schwefel, H. (2002). Evolution Strategies - a comprehensive introduction. *Natural Computing*, *1*, 3–52. doi:10.1023/A:1015059928466

Bhan, A., Galas, D. J., & Dewey, T. G. (2002). A duplication growth model of gene expression networks. *Bioinformatics (Oxford, England)*, *18*(11), 1486–1493. doi:10.1093/bioinformatics/18.11.1486

Biere, A., Cimatti, A., Clarke, E., & Zhu, Y. (1999). *Symbolic model checking without BDDs.* Springer.

Billington, J., Diaz, M., & Rozenberg, G. (Eds.). (1999). *Application of Petri nets to communication networks* (Vol. 1605). Berlin: Springer-Verlag.

Bing, N., & Hoeschele, I. (2005). Genetical genomics analysis of a yeast segregant population for transcription network inference. *Genetics*, *170*(2), 533–542. doi:10.1534/genetics.105.041103

Bittner, M. (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, *406*(6795), 536–540. doi:10.1038/35020115

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., & Hendrix, M. (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, *406*(6795), 536–540. doi:10.1038/35020115

Blake, W. J., Kaern, M., Cantor, C. R., & Collins, J. J. (2003). Noise in eukaryotic gene expression. *Nature*, *422*, 633–637. doi:10.1038/nature01546

Bloem, R., Gabow, H. N., & Somenzi, F. (2006). An algorithm for strongly connected component analysis in n

log n symbolic steps. *Formal Methods in System Design*, *28*(1), 37–56. doi:10.1007/s10703-006-4341-z

Bluthgen, N., Kielbasa, S. M., & Herzel, H. (2005). Inferring combinatorial regulation of transcription in silico. *Nucleic Acids Research*, *33*(1), 272–279. doi:10.1093/nar/gki167

Bollen, K. (1989). *Structural equations with latent variable*. Wiley-Interscience.

Bollobas, B. (1998). Modern graph theory. Springer.

Bolouri, H., & Davidson, E. H. (2003). Transcriptional regulatory cascades in development: Initial rates, not steady state, determine network kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(16), 9371–9376. doi:10.1073/pnas.1533293100

Bonneau, R., Reiss, D. J., Shannon, P., Hood, L., Baliga, N. S., & Thorsson, V. (2006). The inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*(5), R36. doi:10.1186/gb-2006-7-5-r36

Boomsma, D. (1996). Using multivariate genetic modeling to detect pleiotropic quantitative trait loci. *Behavior Genetics*, *26*(2), 161–166. doi:10.1007/BF02359893

Borevitz, J. O., Liang, D., Plouffe, D., Chang, H. S., Zhu, T., & Weigel, D. (2003). Large-scale identification of single-feature polymorphisms in complex genomes. *Genome Research*, *13*(3), 513–523. doi:10.1101/gr.541303

Borisuk, M. T., & Tyson, J. J. (1998). Bifurcation analysis of a model of mitotic control in frog eggs. *Journal of Theoretical Biology*, *195*(1), 69–85. doi:10.1006/jtbi.1998.0781

Bornholdt, S. (2005). Less is more in modeling large genetic networks. *Science*, *310*, 449. doi:10.1126/science.1119959

Boutilier, C., Dean, T., & Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, *11*, 1–94.

Bower, J. M., & Bolouri, H. (Eds.). (2001). *Computational modeling of genetic and biochemical networks*. Cambridge, MA: MIT Press.

Bowers, C. P. (2006). *Simulating evolution with a computational model of embryogeny*. Doctoral Dissertation, The University of Birmingham, UK.

Bozdech, Z., Llinás, M., Pulliam, B. L., Wong, E. D., Zhu, J., & DeRisi, J. L. (2003). The transcriptome of the intraerythrocytic development cycle of plasmodium falciparum . *PLoS Biology*, *1*(1), E5. doi:10.1371/journal.pbio.0000005

Bratsun, D., Volfson, D., Tsimring, L. S., & Hasty, J. (2005). Delay-induced stochastic oscillations in gene regulation. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 14593. doi:10.1073/pnas.0503858102

Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene networks: How to put the function in genomics. *Trends in Biotechnology*, *20*(11), 467–472. doi:10.1016/S0167-7799(02)02053-X

Brazma, A., & Schlitt, T. (2003). Reverse engineering of gene regulatory networks: A finite state linear model. *Genome Biology*, *4*(6). doi:10.1186/gb-2003-4-6-p5

Brazma, A., Jonassen, I., Vilo, J., & Ukkonen, E. (1998). Predicting gene regulatory elements in silico on a genomic scale. *Genome Research*, *8*(11), 1202.

Breitkreutz, B. J., Stark, C., Reguly, T., Boucher, L., Breitkreutz, A., & Livstone, M. (2008). The BioGRID interaction database: 2008 update. *Nucleic Acids Research*, *36*(Database issue), D637. doi:10.1093/nar/gkm1001

Breitling, R., Amtmann, A., & Herzyk, P. (2004). Iterative group analysis (iGA): A simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. *BMC Bioinformatics*, *5*, 34. doi:10.1186/1471-2105-5-34

Brem, R. B., & Kruglyak, L. (2005). The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(5), 1572–1577. doi:10.1073/pnas.0408709102

Brem, R. B., Storey, J. D., Whittle, J., & Kruglyak, L. (2005). Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature*, *436*(7051), 701–703. doi:10.1038/nature03865

Brem, R. B., Yvert, G., Clinton, R., & Kruglyak, L. (2002). Genetic dissection of transcriptional regulation in budding yeast. *Science*, *296*(5568), 752–755. doi:10.1126/science.1069516

Brent, R. (2004). A partnership between biology and engineering. *Nature Biotechnology*, *22*, 1211–1214. doi:10.1038/nbt1004-1211

Briggs, G. E., & Haldane, J. B. S. (1925). A note on the kinetics of enzyme action. *The Biochemical Journal, 19,* 339–339.

Brivanlou, A. H., & Darnell, J. E. Jr. (2002). Signal transduction and the control of gene expression. *Science, 295*(5556), 813–818. doi:10.1126/science.1066355

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., et al. (2000). Graph structure in the Web. Computer Networks, 33(1-6), 309-320.

Brown, C. T. (2008). Computational approaches to finding and analyzing cis-regulatory elements. *Methods in Cell Biology, 87,* 337–365. doi:10.1016/S0091-679X(08)00218-5

Brown, C. T., Rust, A. G., Clarke, P. J. C., Pan, Z., Schilstra, M. J., & Buysscher, T. D. (2002). New computational approaches for analysis of cis-regulatory networks. *Developmental Biology, 246,* 86–102. doi:10.1006/dbio.2002.0619

Brown, P. J., Vanucci, M., & Fearn, T. (2002). Bayes model averaging with selection of regressors. *Journal of the Royal Statistical Society. Series B. Methodological, 64,* 519–536. doi:10.1111/1467-9868.00348

Bryant, R. (1986). Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers, C-35*(8), 677–691. doi:10.1109/TC.1986.1676819

Buck, M. J., & Lieb, J. D. (2004). ChIP-chip: Considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics, 83*(3), 349–360. doi:10.1016/j.ygeno.2003.11.004

Bueno Filho, J. S. S., Gilmour, S. G., & Rosa, G. J. M. (2006). Design of microarray experiments for genetical genomics studies. *Genetics, 174*(2), 945–957. doi:10.1534/genetics.106.057281

Bulashevska, S., & Eils, R. (2005). Inferring genetic regulatory logic from expression data. *Bioinformatics (Oxford, England), 21*(11), 2706–2713. doi:10.1093/bioinformatics/bti388

Bulashevska, S., Adebiyi, E., Brors, B., & Eils, R. (2007). New insights into the genetic regulation of Plasmodium falciparum obtained by Bayesian modeling. *Gene Regulation and Systems Biology, 1,* 117–129.

Burrage, K., Burrage, P. M., Leier, A., et al. (2008). Stochastic delay models for molecular clocks and somite formation. In *Proceedings of SPIE,* 68020Z.

Burrage, K., Hegland, M., MacNamara, S., et al. (2006). A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. In A. N. Langville & W. J. Stewart (Eds.), *Proceedings of the Markov 150th Anniversary Conference* (pp. 21-38). Boson Books.

Bussemaker, H. J., Foat, B. C., & Ward, L. D. (2007). Predictive modeling of genomewide mRNA expression: From modules to molecules. *Annual Review of Biophysics and Biomolecular Structure, 36,* 329–347. doi:10.1146/annurev.biophys.36.040306.132725

Bussemaker, H. J., Li, H., & Siggia, E. D. (2001). Regulatory element detection using correlation with expression. *Nature Genetics, 27,* 167–171. doi:10.1038/84792

Butte, A. J., & Kohane, I. S. (2000). Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing,* 418–429.

Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R., & Kohane, I. S. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences of the United States of America, 97*(22), 12182–12186. doi:10.1073/pnas.220392197

Byrne, J. C., Valen, E., Tang, M. H., Marstrand, T., Winther, O., & da Piedade, I. (2008). JASPAR, the open access database of transcription factor-binding profiles: New content and tools in the 2008 update. *Nucleic Acids Research, 36*(Database issue), D102–D106. doi:10.1093/nar/gkm955

Bystrykh, L., Weersing, E., Dontje, B., Sutton, S., Pletcher, M. T., & Wiltshire, T. (2005). Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics'. *Nature Genetics, 37*(3), 225–232. doi:10.1038/ng1497

Cai, L., Friedman, N., & Xie, X. S. (2006). Stochastic protein expression in individual cells at the single molecule level. *Nature, 440*(7082), 358–362. doi:10.1038/nature04599

Cai, L., Huang, H., Blackshaw, S., Liu, J. S., Cepko, C., & Wong, W. H. (2004). Clustering analysis of SAGE data using a Poisson approach. *Genome Biology, 5*(7), R51. doi:10.1186/gb-2004-5-7-r51

Cai, X. (2007). Exact stochastic simulation of coupled chemical reactions with delays. *The Journal of Chemical Physics*, *126*(12), 124108–124116. doi:10.1063/1.2710253

Camacho, D., de la Fuente, A., & Mendes, P. (2005). The origin of correlations in metabolomic data. *Metabolomics*, *1*(1), 53–63. doi:10.1007/s11306-005-1107-3

Campos, L. M. d., & Castellano, J. G. (2007). Bayesian network learning algorithms using structural restrictions. *International Journal of Approximate Reasoning*, *45*(2), 233–254. doi:10.1016/j.ijar.2006.06.009

Cancho, R. F., & Sole, R. V. (2001). The small-world of human language. *Proceedings. Biological Sciences*, *268*(1482), 2261–2265. doi:10.1098/rspb.2001.1800

Cao, Y., & Petzold, L. (2005). Trapezoidal tau-leaping formula for the stochastic simulation of biochemical systems. In *Foundations of Systems Biology in Engineering*, 149–152.

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2005). Avoiding negative populations in explicit tau leaping. *The Journal of Chemical Physics*, *123*, 054104. doi:10.1063/1.1992473

Cao, Y., Gillespie, D. T., & Petzold, L. R. (2006). Efficient stepsize selection for the tau-leaping method. *The Journal of Chemical Physics*, *124*, 044109. doi:10.1063/1.2159468

Cao, Y., Gillespie, D., & Petzold, L. (2005). The slow-scale stochastic simulation algorithm. *The Journal of Chemical Physics*, 122.

Cao, Y., Li, H., & Petzold, L. (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting system. *The Journal of Chemical Physics*, *121*, 4059–4067. doi:10.1063/1.1778376

Carlborg, O., De Koning, D. J., Manly, K. F., Chesler, E., Williams, R. W., & Haley, C. S. (2005). Methodological aspects of the genetic dissection of gene expression. *Bioinformatics (Oxford, England)*, *21*(10), 2383–2393. doi:10.1093/bioinformatics/bti241

Carlin, B. P., & Chib, S. (1995). Bayesian model choice via Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society. Series B. Methodological*, *57*, 473–484.

Carmi, S., Levanon, E. Y., Havlin, S., & Eisenberg, E. (2006). Connectivity and expression in protein networks: Proteins in a complex are uniformly expressed. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *73*, 031909. doi:10.1103/PhysRevE.73.031909

Carter, D., Chakalova, L., Osborne, C. S., Dai, Y. F., & Fraser, P. (2006). Long-range chromatin regulatory interactions in vivo. *Nature Genetics*, *32*(4), 623–626. doi:10.1038/ng1051

Casella, G., & Berger, R. (2001). *Statistical inference*. Belmont, CA: Duxbury Press.

Castelo, R., & Siebes, A. (2000). Priors on network structures. Biasing the search for Bayesian networks. *International Journal of Approximate Reasoning*, *24*(1), 39–57. doi:10.1016/S0888-613X(99)00041-9

Causton, H. C., Ren, B., Koh, S. S., Harbison, C. T., Kanin, E., & Jennings, E. G. (2001). Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the Cell*, *12*(2), 323–337.

Chabrier, N., & Fages, F. (2003). *Symbolic model checking of biochemical networks.* (. *LNCS*, *2602*, 149–162.

Chang, Y.-H., Wang, Y.-C., & Chen, B.-S. (2006). Identification of transcription factor cooperativity via stochastic system model. *Bioinformatics (Oxford, England)*, *22*(18), 2276–2282. doi:10.1093/bioinformatics/btl380

Chaouiya, C. (2007). Petri net modelling of biological networks. *Briefings in Bioinformatics*, *8*(4), 210–219. doi:10.1093/bib/bbm029

Charlebois, D., Ribeiro, A. S., Lehmussola, A., Lloyd-Price, J., Yli-Harja, O., & Kauffman, S. A. (2008). (accepted). Effects of microarray noise on inference efficiency of a stochastic model of gene networks. *WSEAS Transactions in Biology.*

Chatterjee, A., & Vlachos, D. G. (2006). Multiscale spatial Monte Carlo simulations: Multigriding, computational singular perturbation, and hierarchical stochastic closures. *The Journal of Chemical Physics*, *124*, 064110. doi:10.1063/1.2166380

Chatterjee, A., & Vlachos, D. G. (2006). Temporal acceleration of spatially distributed kinetic Monte Carlo simulations. *Journal of Computational Physics*, *211*(2), 596–615. doi:10.1016/j.jcp.2005.06.004

Chatterjee, A., Vlachos, D. G., & Katsoulakis, M. A. (2005). Binomial distribution based tau-leap accelerated stochastic simulation. *The Journal of Chemical Physics*, *122*, 024112. doi:10.1063/1.1833357

Chen, H. C., Lee, H. C., Lin, T. Y., Li, W. H., & Chen, B. S. (2004). Quantitative characterization of the transcriptional regulatory network in the yeast cell cycle. *Bioinformatics (Oxford, England), 20*, 1914–1927. doi:10.1093/bioinformatics/bth178

Chen, K. C., Calzone, L., Csikasz-Nagy, A., Cross, F., Novak, B., & Tyson, J. J. (2004). Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, *15*(8), 3841–3862. doi:10.1091/mbc.E03-11-0794

Chen, K. C., Wang, T. Y., Tseng, H. H., Huang, C. Y., & Kao, C. Y. (2005). A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae. *Bioinformatics (Oxford, England), 21*, 2883–2890. doi:10.1093/bioinformatics/bti415

Chen, K., & Rajewsky, N. (2007). The evolution of gene regulation by transcription factors and microRNAs. *Nature Reviews. Genetics*, *8*(2), 93–103. doi:10.1038/nrg1990

Chen, K.-C., Wang, T.-Y., Tseng, H.-H., Huang, F., & Kao, C.-Y. (2005). A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae. *Bioinformatics (Oxford, England), 21*(12), 2883–2890. doi:10.1093/bioinformatics/bti415

Chen, L., & Storey, J. D. (2006). Relaxed significance criteria for linkage analysis. *Genetics*, *173*, 2371–2381. doi:10.1534/genetics.105.052506

Chen, L., & Wang, R. (2006). Designing gene regulatory networks with specified functions. *IEEE Transactions on Circuits and Systems*, *53*(11), 2444–2450. doi:10.1109/TCSI.2006.883880

Chen, M., & Hofestädt, R. (2003). Quantitative Petri net model of gene regulated metabolic networks in the cell. In *Silico Biology*, *3*.

Chen, M.-H., & Ibrahim, J. G. (2003). Conjugate priors for generalized linear models. *Statistica Sinica*, *13*, 461–476.

Chen, T., He, H. L., & Church, G. M. (1999). Modeling gene expression with differential equations. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *4*, 29–40.

Chen, Y., Zhu, J., Lum, P. Y., Yang, X., Pinto, S., Macneil, D. J., et al. (2008). Variations in DNA elucidate molecular networks that cause disease. *Nature*.

Cheng, J., Bell, D. A., & Liu, W. (1997). Learning belief networks from data: An information theory based approach. *The Sixth ACM International Conference on Information and Knowledge Management* (pp. 325-331).

Chesler, E. J., Lu, L., Shou, S., Qu, Y., Gu, J., & Wang, J. (2005). Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, *37*(3), 233–242. doi:10.1038/ng1518

Cheung, V. G., Jen, K. Y., Weber, T., Morley, M., Devlin, J. L., & Ewens, K. G. (2003). Genetics of quantitative variation in human gene expression. *Cold Spring Harbor Symposia on Quantitative Biology*, *68*, 403–407.

Cheung, V. G., Spielman, R. S., Ewens, K. G., Weber, T. M., Morley, M., & Burdick, J. T. (2005). Mapping determinants of human gene expression by regional and genome-wide association. *Nature*, *437*(7063), 1365–1369. doi:10.1038/nature04244

Chickering, D. M. (2002). Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, *2*, 445–498. doi:10.1162/153244302760200696

Chickering, D. M., Geiger, D., & Heckerman, D. (1994). Learning Bayesian networks is NP-hard. (Tech. Rep. MSR-TR-94-17). Microsoft Research.

Chien, C., Bartel, P. L., Sternglanz, R., & Fields, S. (1991). The two-hybrid system: A method to identify and clone genes for proteins that interact with a protein of interest. *Proceedings of the National Academy of Sciences of the United States of America*, *88*(21), 9578–9582. doi:10.1073/pnas.88.21.9578

Chipman, H. (1996). Bayesian variable selection with related predictors. *The Canadian Journal of Statistics*, *24*, 17–36. doi:10.2307/3315687

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., & Wodicka, L. (1998). A genome wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, *2*, 65–73. doi:10.1016/S1097-2765(00)80114-8

Christensen, C., Gupta, A., Maranas, C. D., & Albert, R. (2007). Inference and graph-theoretical analysis of Bacillus Subtilis gene regulatory networks. *Physica A*, *373*, 796–810. doi:10.1016/j.physa.2006.04.118

Chu, D. (2007). Evolving genetic regulatory networks for systems biology. In *Proceedings of the Congress on Evolutionary Computation 2007*, Singapore (pp. 875-882).

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., & Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science*, *282*, 699–705. doi:10.1126/science.282.5389.699

Chu, T. (2003). Learning from SAGE data. Unpublished doctoral dissertation, Carnegie Mellon University.

Chu, T. (2004). Limitations of statistical learning from gene expression data. *Interface 2004: Computational Biology and Bioinformatics.*

Chu, T., Glymour, C., Scheines, R., & Spirtes, P. (2003). A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays. *Bioinformatics (Oxford, England)*, *19*(9), 1147–1152. doi:10.1093/bioinformatics/btg011

Chung, K. L., & Williams, R. J. (1990). *Introduction to stochastic integration.* Boston: Birkhäuser.

Chung, T.-H., Brun, M., & Kim, S. (2006). Quantization of global gene expression data. *5th International Conference on Machine Learning and Applications* (ICMLA'06) (pp. 187–192).

Churchill, G. A., & Doerge, R. W. (1994). Empirical threshold values for quantitative trait mapping. *Genetics*, *138*(3), 963–971.

Cimatti, A., Clarke, E., Giunchiglia, E., Giunchiglia, F., Pistore, M., Roveri, M., et al. (2002). NuSMV version 2: An opensource tool for symbolic model checking. In *Proc. International Conference on Computer-Aided Verification (CAV 2002)* (Vol. 2404 of LNCS), Copenhagen, Denmark. Springer.

Clamp, M., Fry, B., Kamal, M., Xie, X., Cuff, J., & Lin, M. F. (2007). Distinguishing protein-coding and noncoding genes in the human genome. *Proceedings of the National Academy of Sciences of the United States of America*, *104*, 19428–19433. doi:10.1073/pnas.0709013104

Clarke, E. M., & Emerson, E. A. (1981). Characterizing properties of parallel programs as fixpoints. In *Seventh International Colloqium on Automata, Languages, and Programming* (Vol. 85 of LNCS).

Claverie, J. M. (2005). Fewer genes, more noncoding RNA. *Science*, *309*, 5740. doi:10.1126/science.1116800

Climescu-Haulica, A., & Quirk, M. D. (2007). A stochastic differential equation model for transcriptional regulatory networks. *BMC Bioinformatics*, *8*(Suppl 5), S4. doi:10.1186/1471-2105-8-S5-S4

Clyde, M. (1999). Bayesian model averaging and model search strategies (with discussion). In J. Bernardo, J. Berger, A. Dawid & A. Smith (Eds.), *Bayesian statistics*, 6. Oxford: Clarendon Press.

Clyde, M., DeSimone, H., & Parmigiani, G. (1996). Prediction via orthogonalized model mixing. *Journal of the American Statistical Association*, *91*, 1197–1208. doi:10.2307/2291738

Codd, E. F. (1968). *Cellular automata*. New York: Academic Press.

Cohen, P. (2002). The origins of protein phosphorylation. *Nature Cell Biology*, *4*(5), 127–130. doi:10.1038/ncb0502-e127

Cokus, S. J., Haynor, D., Gronbech-Jensen, N., & Pellegrini, M. (2006). Modelling the network of cell cycle transcription factors in the yeast *Saccharomyces cerevisiae*. *BMC Bioinformatics*, *7*(381).

Collins, F. S., Green, E. D., Guttmacher, A. E., & Guyer, M. S. (2003). A vision for the future of genomics research. *Nature*, *422*(6934), 835–847. doi:10.1038/nature01626

Comuzzie, A. G., Mahaney, M. C., Almasy, L., Dyer, T. D., & Blangero, J. (1997). Exploiting pleiotropy to map genes for oligogenic phenotypes using extended pedigree data. *Genetic Epidemiology*, *14*(6), 975–980. doi:10.1002/(SICI)1098-2272(1997)14:6<975::AID-GEPI69>3.0.CO;2-I

Conlon, E. M., Liu, X. S., Lieb, J. D., & Liu, J. S. (2003). Integrating regulatory motif discovery and genomewide expression analysis. *Proceedings of the National Academy of Sciences of the United States of America*, *100*, 3339–3344. doi:10.1073/pnas.0630591100

Cooper, G. F. (1997). A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, *1*(2), 203–224. doi:10.1023/A:1009787925236

Cooper, G. F., & Herskovits, E. H. (1992). The induction of probabilistic networks from data. *Machine Learning*, *9*(4), 309–347.

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*(4), 309–347.

Cornish-Bowden, A., Cardenas, M. L., Letelier, J. C., & Soto-Andrade, J. (2007). Beyond reductionism: Metabolic circularity as a guiding vision for a real biology of systems. *Proteomics, 7*(6), 839–845. doi:10.1002/pmic.200600431

Costa, F. (2007). Noncoding RNAs: New players in eukaryotic biology. *Gene, 357*(2), 83–94. doi:10.1016/j.gene.2005.06.019

Covert, M. W., Knight, E. M., Reed, J. L., Herrgard, M. J., & Palsson, B. O. (2004). Integrating high-throughput and computational data elucidates bacterial networks. *Nature, 429*, 92–96. doi:10.1038/nature02456

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). Probabilistic networks and expert systems. New York: Springer-Verlag.

Crick, F. (1970). Central dogma of molecular biology. *Nature, 227*, 561–563. doi:10.1038/227561a0

Cui, X., Xu, J., Asghar, R., Condamine, P., Svensson, J. T., & Wanamaker, S. (2005). Detecting single-feature polymorphisms using oligonucleotide arrays and robustified projection pursuit. *Bioinformatics (Oxford, England), 21*(20), 3852–3858. doi:10.1093/bioinformatics/bti640

D'haeseleer, P. (2005). How does gene expression clustering work? *Nature Biotechnology, 23*(12), 1499–1501. doi:10.1038/nbt1205-1499

D'Haeseleer, P., Liang, S., & Somogyi, R. (2000). Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics (Oxford, England), 16*(8), 707–726. doi:10.1093/bioinformatics/16.8.707

D'haeseleer. P. (2000). *Reconstructing gene networks from large scale gene expression data*. Unpublished doctoral dissertation, University of New Mexico, USA.

D'haeseleer. P., Wen, X., Fuhrman, S., & Somogyi, R. (1999). Linear modeling of mRNA expression levels during CNS development and injury. In R. B. Altman, A. K. Dunker, L. Hunter, T. E. Klein & K. Lauderdaule (Eds.), *Pacific Symposium on Biocomputing*, 4, 41–52. Singapore: World Scientific Publishing Co.

Danks, D., & Glymour, C. (2002). Linearity properties of Bayes nets with binary variables. In J. Breese & D. Koller (Eds.), *Uncertainty in artificial intelligence: Proceedings of the 17th conference (UAI-2001)* (pp. 98-104). San Francisco: Morgan Kaufmann.

Danks, D., Glymour, C., & Spirtes, P. (2003). The computational and experimental complexity of gene perturbations for regulatory network search. In W. H. Hsu, R. Joehanes & C. D. Page (Eds.), *Proceedings of IJCAI-2003 workshop on learning graphical models for computational genomics* (pp. 22-31).

Darabos, C., Giacobini, M., & Tomassini, M. (2007). Semi-synchronous activation in scale-free Boolean networks. In F. Almeida, E. Costa, et al. (Eds.), *Advances in Artificial Life, 9th European Conference, ECAL2007* (LNAI, pp. 976-985), Heidelberg. Springer-Verlag.

Darvasi, A. (1998). Experimental strategies for the genetic dissection of complex traits in animal models. *Nature Genetics, 18*(1), 19–24. doi:10.1038/ng0198-19

Datta, A., Choudhary, A., Bittner, M., & Dougherty, E. (2004). External control in Markovian genetic regulatory networks: The imperfect information case. *Bioinformatics (Oxford, England), 20*(6), 924–930. doi:10.1093/bioinformatics/bth008

Davenport, R., White, G., Landick, R., & Bustamante, C. (2000). Single-molecule study of transcriptional pausing and arrest by E. coli rna polymerase. *Science, 287*, 2497–2500. doi:10.1126/science.287.5462.2497

David, R., & Alla, H. (2005). *Discrete, continous, and hybrid Petri nets.* Berlin: Springer-Verlag.

Davidich, M., & Bornholdt, S. (2007). (in press). Boolean network model predicts cell cycle sequence of fission yeast. *PLoS ONE*.

Davidson, E. H. (2002). A genomic regulatory network for development. *Science, 295*, 1669–1678. doi:10.1126/science.1069883

Davidson, E. H., Rast, J. P., Oliveri, P., Ransick, A., Calestani, C., & Yuh, C.-H. (2002). A genomic regulatory network for development. *Science, 295*(5560), 1669–1678. doi:10.1126/science.1069883

Davis, D. D., Dulbecco, R., Eisen, H. N., & Ginsberg, H. S. (Eds.). (1980). *Microbiology*. Philadelphia, PA: Harper & Row.

De Hoon, M. J. L., Ott, S., Imoto, S., & Miyano, S. (2003, September 27-30). Validation of noisy dynamical system models of gene regulation inferred from time-course gene expression data at arbitrary time intervals. *Poster Proceedings of the European Conference on Computational Biology (ECCB 2003),* Paris, France.

de Hoon, S., Imoto, K., & Kobayashi, N. Ogasawara, & Miyano, S. (2003). Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. *Pacific Symposium on Computation Biology*, 8, 17-28.

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*(1), 67–103. doi:10.1089/10665270252833208

De Jong, H., & Page, M. (2000). Qualitative simulation of large and complex genetic regulatory systems. In W. Horn (Ed.), *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI2000)* (pp. 191-195). Berlin: IOS press.

De Jong, H., Geiselmann, J., Hernandez, C., & Page, M. (2003). Genetic network analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics (Oxford, England), 19*(3), 336–344. doi:10.1093/bioinformatics/btf851

De Jong, H., Gouze, J.-L., Hernandez, C., Page, M., Sari, T., & Geiselmann, J. (2004). Qualitative simulation of genetic regulatory networks using piecewise linear models. *ull. Math. Biol., 66*(2), 301–340. doi:10.1016/j.bulm.2003.08.010

de la Fuente, A., & Mendes, P. (2002). Quantifying gene networks with regulatory strengths. *Molecular Biology Reports*, *29*(1-2), 73–77. doi:10.1023/A:1020310504986

de la Fuente, A., Bing, N., Hoeschele, I., & Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics (Oxford, England), 20*, 3565–3574. doi:10.1093/bioinformatics/bth445

de la Fuente, A., Brazhnik, P., & Mendes, P. (2001). A quantitative method for reverse engineering gene networks from microarray experiments using regulatory strengths. Paper presented at the 2nd Int. Conf. Syst. Biol., California Institute of Technology, Pasadena, CA.

de la Fuente, A., Brazhnik, P., & Mendes, P. (2002). Linking the genes: Inferring quantitative gene networks from microarray data. *Trends in Genetics*, *18*(8), 395–398. doi:10.1016/S0168-9525(02)02692-6

de la Fuente, A., Brazhnik, P., & Mendes, P. (2004). Regulatory strength analysis for inferring gene networks. In K. B. N. & W. H. V. (Eds.), Metabolic engineering in the post genomic era (pp. 107-137). Wymondham, UK: Horizon Bioscience.

de Micheli, G. (1994). Synthesis and optimization of digital circuits. McGraw-Hill.

Deans, T. L., Cantor, C. R., & Collins, J. J. (2007). A tunable genetic switch based on RNAi and repressor proteins for regulating gene expression in mammalian cells. *Cell, 130*(2), 363–372. doi:10.1016/j.cell.2007.05.045

DeCook, R., Lall, S., Nettleton, D., & Howell, S. H. (2006). Genetic regulation of gene expression during shoot development in Arabidopsis. *Genetics, 172*(2), 1155–1164. doi:10.1534/genetics.105.042275

Dejana, E., Taddei, A., & Randi, A. M. (2007). Foxs and Ets in the transcriptional regulation of endothelial cell differentiation and angiogenesis. *Biochimica et Biophysica Acta, 1775*(2), 298–312.

Dellaportas, P., Forster, J. J., & Ntzoufras, I. (2000). *Bayesian variable selection using the Gibbs sampler.* In D. K. Dey, S. Ghosh & B. Mallick (Eds.), *Generalized linear models: A Bayesian perspective* (pp. 271-286). New York: Marcel Dekker.

Dellaportas, P., Forster, J. J., & Ntzoufras, I. (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing, 12*, 27–36. doi:10.1023/A:1013164120801

Demeter, J., Beauheim, C., Gollub, J., Hernandez-Boussard, T., Jin, H., & Maier, D. (2007). The Stanford Microarray Database: implementation of new analysis tools and open source release of software. *Nucleic Acids Research, 35*(Suppl 1), D766–D770. doi:10.1093/nar/gkl1019

Dempster, A. P. (1972). Covariance selection. *Biometrics, 28*, 157–175. doi:10.2307/2528966

DeRisi, J. L., Iyer, V. R., & Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science, 278*(5338), 680–686. doi:10.1126/science.278.5338.680

Derrida, B., & Pomeau, Y. (1986). Random networks of automata: A simple annealed approximation. *Europhysics Letters, 1*(2), 45–49. doi:10.1209/0295-5075/1/2/001

Dewey, G. T., & Galas, D. J. (2002). In *Eurekah bioscience collection. Landes biosciences.*

Dewey, T. G., & Galas, D. J. (2001). Dynamic models of gene expression and classification. *Functional & Integrative Genomics, 1*(4), 269–278. doi:10.1007/s101420000035

di Bernardo, D., Thompson, M. J., Gardner, T. S., Chobot, S. E., Eastwood, E. L., & Wojtovich, A. P. (2005). Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nature Biotechnology, 23*(3), 377–383. doi:10.1038/nbt1075

Di Camillo, B., Sanchez-Cabo, F., Toffolo, G., Nair, S. K., Trajanoski, Z., & Cobelli, C. (2005, December 1). A quantization method based on threshold optimization for microarray short time series. *BMC Bioinformatics, 6*(Suppl 4), S11. doi:10.1186/1471-2105-6-S4-S11

Dobrin, R., Beg, Q. K., Barabási, A.-L., & Oltvai, Z. N. (2004). Aggregation of topological motifs in the Escherichia coli transcriptional regulatory network. *BMC Bioinformatics, 5*, 10. doi:10.1186/1471-2105-5-10

Doerge, R. W. (2002). Mapping and analysis of quantitative trait loci in experimental populations. *Nature Reviews. Genetics, 3*(1), 43–52. doi:10.1038/nrg703

Doerge, R. W., Zeng, Z.-B., & Weir, B. S. (1997). Statistical issues in the search for genes affecting quantitative traits in experimental populations. *Statistical Science, 12*, 195–219. doi:10.1214/ss/1030037909

Doi, A., Masao Nagasaki, M., Matsuno, H., & Satoru Miyano, S. (2003). Genomic object net II: How to model biopathways with hybrid functional Petri net with extension. *Applied Bioinformatics, 2*(3), 185–188.

Dojer, N., Gambin, A., Mizera, A., Wilczynski, B., & Tiuryn, J. (2006). Applying dynamic Bayesian networks to perturbed gene expression data. *BMC Bioinformatics, 7*(249).

Dorogovtsev, S. N., & Mendes, J. F. F. (2003). *Evolution of networks: From biological networks to the Internet and WWW.* Oxford: Oxford University Press.

Doss, S., Schadt, E. E., Drake, T. A., & Lusis, A. J. (2005). Cis-acting expression quantitative trait loci in mice. *Genome Research, 15*(5), 681–691. doi:10.1101/gr.3216905

Dougherty, E. R., & Braga-Neto, U. (2006). Epistemology of computational biology: Mathematical models and experimental prediction as the basis of their validity. *Journal of Computational Biology, 14*(1), 65–90.

Dougherty, E. R., Datta, A., & Sima, C. (2005). Research issues in genomic signal processing. *IEEE Signal Processing Magazine, 22*(6), 46–68. doi:10.1109/MSP.2005.1550189

Dougherty, E. R., Kim, S., & Chen, Y. (2000). Coefficient of determination in nonlinear signal processing. *Signal Processing, 80*, 2219–2235. doi:10.1016/S0165-1684(00)00079-7

Dougherty, J., & Ivanov, I. (in press). Reduction cost for Boolean networks with perturbation.

Draper, N. R., & Smith, H. (1998). *Applied regression analysis*, 3rd ed. New York: Wiley.

DREAM. Dialogue on Reverse-Engineering Assessment and Methods. (2007). Retrieved in January 2009, from http://wiki.c2b2.columbia.edu/dream/index.php/The_DREAM_Project

DREAM. Dialogue on Reverse-Engineering Assessment and Methods. (2008). Descriptions of the challenges. Retrieved in January 2009, from http://wiki.c2b2.columbia.edu/dream/index.php/The_DREAM3_Challenges

Dubois, D., Nguyen, H. T., Prade, H., & Sugeno, M. (1998). Introduction: The real contribution of fuzzy systems. In H. T. Nguyen & M. Sugeno (Eds.), *Fuzzy systems: Modeling and control* (pp. 1-17). Kluwer.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. John Wiley & Sons, 2nd edition.

Duncan, S. A., Navas, M. A., Dufort, D., Rossant, J., & Stoffel, M. (1998). Regulation of a transcription factor network required for differentiation and metabolism. *Science, 281*(5377), 692–695. doi:10.1126/science.281.5377.692

Eads, B., Cash, A., Bogart, K., Costello, J., & Andrews, J. (2006). Troubleshooting microarray hybridizations. *Methods in Enzymology, 411*, 34–39. doi:10.1016/S0076-6879(06)11003-4

Eberhardt, F. (2007). Causation and intervention. Unpublished doctoral dissertation, Carnegie Mellon University.

Edwards, D. (1995). Introduction to graphical modelling. Springer-Verlag.

Edwards, J. S., Ibarra, R. U., & Palsson, B. O. (2001). In silico predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. *Nature Biotechnology, 19*, 125–130. doi:10.1038/84379

Edwards, J. S., P. B. O. (2000). The Escherichia coli MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities. *Proceedings of the National Academy of*

Sciences of the United States of America, 97(10), 5528–5533. doi:10.1073/pnas.97.10.5528

Edwards, R., & Glass, L. (2006). A calculus for relating the dynamics and structure of complex biological networks. In R. S. Berry & J. Jortner (Eds.), *Advances in chemical physics* (pp. 151-178). New York: J. Wiley and Sons.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *Annals of Statistics, 32*, 407–499. doi:10.1214/009053604000000067

Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. *Proceedings of the 4th European Conference on Artificial Life* (pp. 205-213).

Eisen, M. B., Spelmman, P. T., Brown, P. O., & Botstein, O. (1998). Cluster analysis and display of genomewide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(25), 14863–14868. doi:10.1073/pnas.95.25.14863

Elf, J., & Ehrenberg, M. (2004). Spontaneous separation of bistable biochemical systems into spatial domains of opposite phases. *Systems Biology, 2*, 230. doi:10.1049/sb:20045021

Elf, J., Doncic, A., & Ehrenberg, M. (2003). Mesoscopic reaction-diffusion in intracellular signaling. *Proceedings of the Society for Photo-Instrumentation Engineers, 5110*, 114–124. doi:10.1117/12.497009

Elnitski, L., Jin, V. X., Farnham, P. J., & Jones, S. J. M. (2006). Locating mammalian transcription factor binding sites: A survey of computational and experimental techniques. *Genome Research, 16*(12), 1455–1464. doi:10.1101/gr.4140006

Elowitz, M. B., & Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, *403*, 335–338. doi:10.1038/35002125

Elowitz, M. B., Levine, A. J., Siggia, E. D., & Swain, P. S. (2002). Stochastic gene expression in a single cell. *Science*, *297*(5584), 1183–1186. doi:10.1126/science.1070919

Elsasser, S., & Finley, D. (2005). Delivery of ubiquitinated substrates to protein-unfolding machines. *Nature Cell Biology*, *7*(8), 742–749. doi:10.1038/ncb0805-742

Emilsson, V., Thorleifsson, G., Zhang, B., Leonardson, A. S., Zink, F., Zhu, J., et al. (2008). Genetics of gene expression and its effect on disease. *Nature*.

Endesfelder, S., Bucher, S., Kliche, A., Reszka, R., & Speer, A. (2003). Transfection of normal primary human skeletal myoblasts with p21 and p57 antisense oligonucleotides to improve their proliferation: A first step towards an alternative molecular therapy approach of Duchenne muscular dystrophy. *Journal of Molecular Medicine, 81*, 355–362.

Endesfelder, S., Kliche, A., Lochmüller, H., von Moers, A., & Speer, A. (2005). Antisense oligonucleotides and short interfering RNAs silencing the cyclin-dependent kinase inhibitor p21 improve proliferation of Duchenne muscular dystrophy patients' primary skeletal myoblasts. *Journal of Molecular Medicine, 83*, 64–71. doi:10.1007/s00109-004-0607-3

Erdös, P., & Rényi, A. (1959). On random graphs. *Publicationes Mathematicae (Debrecen), 6*, 290–297.

Eriksen, K. A., & Hornquist, M. (2001). Scale-free growing networks imply linear preferential attachment. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics, 65*(1 pt 2).

Esparza, J., & Nielsen, M. (1994). Decidabilty issues for Petri nets. *Journal of Information Processing and Cybernetics, EIK, 30*, 143–160.

Fages, F., Soliman, S., & Chabrier-Rivier, N. (2004). Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM. *J Biological Physics and Chemistry, 4*(2), 64–73. doi:10.4024/2040402.jbpc.04.02

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., & Cottarel, G. (2007). Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biology, 5*(1), e8. doi:10.1371/journal.pbio.0050008

Fasken, M. B., & Corbett, A. H. (2005). Process or perish: Quality control in mRNA biogenesis. *Nature Structural & Molecular Biology, 12*, 482–488. doi:10.1038/nsmb945

Federici, D. (2004). Using embryonic stages to increase evolvability of development. In J. Miller (Ed.), *Proceedings of the Workshop on Regeneration and Learning in Developmental Systems WORLDS 2004*.

Fiehn, O., & Weckwerth, W. (2003). Deciphering metabolic networks. *European Journal of Biochemistry, 270*(4), 579–588. doi:10.1046/j.1432-1033.2003.03427.x

Filkov, V., Skiena, S., & Zhi, J. (2002). Analysis techniques for microarray time-series data. *Journal of Computational Biology, 9*, 317–330. doi:10.1089/10665270252935485

Finney, A., & Hucka, M. (2003). Systems biology markup language (SBML) level 2: Structures and facilities for model definitions. Retrieved from http://www.sbml.org/

Fisher, F. M. (1970). A correspondence principle for simultanious equation models. *Econometrica*, *38*(1), 73–92. doi:10.2307/1909242

Fisher, R. A. (1954). *Statistical methods for research workers* (12th edition). Edinburgh, UK.

Fisk, D. G., Ball, C. A., Dolinski, K., Engel, S. R., Hong, E. L., & Issel-Tarver, L. (2006). Saccharomyces cerevisiae S288C genome annotation: A working hypothesis. [from http://www.yeastgenome.org/]. *Yeast (Chichester, England)*, *23*(12), 857–865. Retrieved in December 2007. doi:10.1002/yea.1400

Fiúza, U.-M., & Arias, A. M. (2007). Cell and molecular biology of Notch. *The Journal of Endocrinology*, *194*, 459–474. doi:10.1677/JOE-07-0242

Fodor, S. P., Rava, R. P., Huang, X. C., Pease, A. C., Holmes, C. P., & Adams, C. L. (1993). Multiplexed biochemical assays with biological chips. *Nature*, *364*, 555–556. doi:10.1038/364555a0

Fogel, D. B. (1995). *Evolutionary computation. Towards a new philosophy of machine intelligence*. IEEE Press.

Forbes, N. (2000). Life as it could be: Alife attempts to simulate evolution. *Intelligent Systems and Their Applications, IEEE*, *15*(6), 2–7.

Foss, E. J., Radulovic, D., Shaffer, S. A., Ruderfer, D. M., Bedalov, A., & Goodlett, D. R. (2007). Genetic basis of proteome variation in yeast. *Nature Genetics*, *39*(11), 1369–1375. doi:10.1038/ng.2007.22

Fowlkes, C. C., Hendriks, C. L., Keränen, S. V., Weber, G. H., Rübel, O., & Huang, M. Y. (2007). A quantitative spatiotemporal atlas of gene expression in the Drosophila blastoderm. *Cell*, *133*(2), 364–374. doi:10.1016/j.cell.2008.01.053

Fraga, M. F., & Esteller, M. (2005). Towards the human cancer epigenome: A first draft of histone modifications. *Cell Cycle (Georgetown, Tex.)*, *4*(10), 1377–1381.

Francois, P., & Hakim, V. (2004). Design of genetic networks with specified functions by evolution in silico. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(2), 580–585. doi:10.1073/pnas.0304532101

Franke, L., Bakel, H., Fokkens, L., de Jong, E. D., Egmont-Petersen, M., & Wijmenga, C. (2006). Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *American Journal of Human Genetics*, *78*(6), 1011–1025. doi:10.1086/504300

Frenster, J. H., & Hovsepian, J. A. (2002). RNA feedback mechanisms during eukaryotic gene regulation. In *Northwest symposium on systems biology* (p. 15).

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Lasso and elastic-net regularized generalized linear models. Retrieved from http://www-stat.stanford.edu/~hastie/Papers/glmnet.pdf

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, *303*(5659), 799–805. doi:10.1126/science.1094068

Friedman, N., & Koller, D. (2003). Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, *50*(1-2), 95–125. doi:10.1023/A:1020249912095

Friedman, N., Goldszmidt, M., & Wyner, A. (1999). *Data analysis with Bayesian networks: A bootstrap approach*.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3-4), 601–620. doi:10.1089/106652700750050961

Friedman, N., Murphy, K., & Russell, S. (1998). Learning the structure of dynamical probabilistic networks. *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence* (pp. 139-147). San Francisco: Morgan Kaufmann Publishers.

Friedman, N., Nachman, I., & Pe'er, D. (1999). Learning Bayesian network structure from massive datasets: The 'sparse candidate' algorithm. In K. Laskey & H. Prade (Eds.), *Proceedings of the 15th international conference on uncertainty in artificial intelligence* (pp. 206-215). San Francisco, CA: Morgan Kaufmann.

Friedman, N., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Recomb 2000*, Tokyo.

Frith, M. C., Spouge, J. L., Hansen, U., & Weng, Z. (2002). Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences.

Nucleic Acids Research, *30*(14), 3214–3224. doi:10.1093/nar/gkf438

Fu, J., & Jansen, R. C. (2006). Optimal design and analysis of genetic studies on gene expression. *Genetics*, *172*(3), 1993–1999. doi:10.1534/genetics.105.047001

Fung, E., Wong, W. W., Suen, J. K., Bulter, T., Lee, S., & Liao, J. C. (2005). A synthetic gene-metabolic oscillator. *Nature*, *435*, 118–122. doi:10.1038/nature03508

Galperin, M. Y. (2004). Bacterial signal transduction network in a genomic perspective. *Environmental Microbiology*, *6*(6), 552–567. doi:10.1111/j.1462-2920.2004.00633.x

Gama-Castro, S., Jimenez-Jacinto, V., Peralta-Gil, M., Santos-Zavaleta, A., Penaloza-Spinola, M. I., & Contreras-Moreira, B. (2008). RegulonDB (version 6.0): Gene regulation model of Escherichia coli K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic Acids Research*, *36*(Database issue), D120–D124. doi:10.1093/nar/gkm994

Gamerman, D., & Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic simulation for Bayesian inference*. Chapman & Hall/CRC.

Gantmacher, F. R. (1960). The theory of matrices (Vol. II). New York: Chelsea Publishing Company.

Gao, J., Li, W. X., Feng, S. Q., Yuan, Y. S., Wan, D. F., Han, W., & Yu, Y. (2008). (in press). A protein-protein interaction network of transcription factors acting during liver cell proliferation. *Genomics*.

Gardiner, C. W. (2004). *Handbook of stochastic methods: For physics, chemistry, and the natural sciences*. Springer, 3rd edition.

Gardner, T. S., & Faith, J. J. (2005). Reverse-engineering transcription control networks. *Physics of Life Reviews*, *2*(1), 65–88. doi:10.1016/j.plrev.2005.01.001

Gardner, T. S., Cantor, C. R., & Collins, J. J. (2000). Construction of a genetic toggle switch in Escherichia coli. *Nature*, *403*, 339–342. doi:10.1038/35002131

Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *301*(5629), 102–105. doi:10.1126/science.1081900

Garg, A., Xenarios, I., Mendonza, L., & DeMicheli, G. (2007). An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments.) . *LNCS*, *4453*, 62–76.

Gasch, A. P., Spellman, P. T., Kao, C. M., Cramel-Harel, O., Eisen, M. B., & Storz, G. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, *11*, 4241–4257.

Gat-Viks, I., Tanay, A., Raijman, D., & Shamir, R. (2006). A probabilistic methodology for integrating knowledge and experiments on biological networks. *Journal of Computational Biology*, *13*(2), 165–181. doi:10.1089/cmb.2006.13.165

Gavin, A.-C., Bosche, M., Krause, R., & Grandi, P. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, *415*(6868), 141–147. doi:10.1038/415141a

Geard, N. (2004). Modeling gene regulatory networks: Systems biology to complex systems. (Tech. Rep. ACCS Draft). Australia: The University of Queensland.

Geier, F., Timmer, J., & Fleck, C. (2007). Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge. *BMC Systems Biology*, *1*(1), 11. doi:10.1186/1752-0509-1-11

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–472. doi:10.1214/ss/1177011136

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. In C. Chatfield, M. Tanner & J. Zidek (Eds.), *Texts in statistical science series*, 2nd edition. Chapman & Hall/CRC.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741. doi:10.1109/TPAMI.1984.4767596

Genovese, C., & Wasserman, L. (2001). False discovery rates (Tech. Rep. 762). Carnegie Mellon University: Department of Statistics.

Genrich, H., Küffner, R., & Voss, K. (2001). Executable Petri net models for the analysis of metabolic pathways. *International Journal on Software Tools for Technoloy Transfer*, *4*, 1–11.

George, E. I., & McCulloch, R. E. (1993). Stochastic search variable selection. In W. R. Gilks, S. Richardson & D. J.

Spiegelhalter (Eds.), *Markov Chain Monte Carlo in Practice* (pp. 203-214). London: Chapman and Hall.

Gershenson, C. (2004). Updating schemes in random Boolean networks: Do they really matter? In J. Pollack (Ed.), *Artificial Life IX Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (pp. 238-243). MIT Press.

Gersho, A., & Gray, R., M. (1992). *Vector quantization and signal compression.* The Kluwer International Series in Engineering and Computer Science.

Gevaert, O., Van Vooren, S., & De Moor, B. (2007). A framework for elucidating regulatory networks based on prior information and expression data. *Annals of the New York Academy of Sciences, 1115*(1), 240–248. doi:10.1196/annals.1407.002

Geva-Zatorsky, N., Rosenfeld, N., Itzkovitz, S., Milo, R., Sigal, A., & Dekel, E. (2006). Oscillations and variability in the p53 system . *Molecular Systems Biology, 2*. doi:. doi:10.1038/msb4100068

Ghaffari, N., Ivanov, I., & Dougherty, E. R. (in press). Reduction mappings and control policies for intervention in Boolean networks.

Ghazalpour, A., Doss, S., Zhang, B., Wang, S., Plaisier, C., & Castellanos, R. (2006). Integrating genetic and network analysis to characterize genes related to mouse weight. *PLOS Genetics, 2*(8), e130. doi:10.1371/journal.pgen.0020130

Giacobini, M., Tomassini, M., De Los Rios, P., & Pestelacci, E. (2006). Dynamics of scale-free semi-synchronous Boolean networks. In L. M. Rocha, et al. (Eds.), *Artificial Life X* (pp. 1-7). Cambridge, MA: The MIT Press.

Gibson, M. A., & Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A, 104*, 1876–1889. doi:10.1021/jp993732q

Gilbert, E. N. (1959)... *Annals of Mathematical Statistics, 30*, 1141. doi:10.1214/aoms/1177706098

Gilbert, S. F. (Ed.). (2003). *Developmental biology, seventh edition.* Sinaur Associates, Inc.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1993). *Markov Chain Monte Carlo in practice*. London: Chapman & Hall.

Gillespie, D. (2000). The chemical langevin equation. *The Journal of Chemical Physics, 113*(1). doi:10.1063/1.481811

Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics, 22*, 403–434. doi:10.1016/0021-9991(76)90041-3

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry, 81*(25), 2340–2361. doi:10.1021/j100540a008

Gillespie, D. T. (1992). *Markov processes an introduction for physical scientists.* Academic Press, Inc.

Gillespie, D. T. (1992). A rigorous derivation of the chemical master equation. *Physica A, 188*, 404–425. doi:10.1016/0378-4371(92)90283-V

Gillespie, D. T. (2000). The chemical Langevin equation. *The Journal of Chemical Physics, 113*, 297–306. doi:10.1063/1.481811

Gillespie, D. T. (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *The Journal of Chemical Physics, 115*(4), 1716–1733. doi:10.1063/1.1378322

Gillespie, D. T. (2005). *Handbook of materials modeling* (pp. 1735–1752). Springer.

Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry, 58*(1), 35–55. doi:10.1146/annurev.physchem.58.032806.104637

Gillespie, D. T., & Petzold, L. R. (2003). Improved leap-size selection for accelerated stochastic simulation. *The Journal of Chemical Physics, 119*(16), 8229–8234. doi:10.1063/1.1613254

Ginsberg, S. D., Elarova, I., Ruben, M., Tan, F., Counts, S. E., & Eberwine, J. H. (2004). Single-cell gene expression analysis: Implications for neurodegenerative and neuropsychiatric disorders. *Neurochemical Research, 29*, 1053–1064. doi:10.1023/B:NERE.0000023593.77052.f7

Giorgetti, N. (2005). Matlab MEX interface for the CPLEX callable library. Retrieved in December 2007, from http://www.dii.unisi.it/~hybrid/tools/mex/downloads.html CPLEX version 10 retrieved from http://www.ilog.com/ (commercial software)

Giudicelli, F., & Lewis, J. (2004). The vertebrate segmentation clock. *Current Opinion in Genetics & Development*, *14*(4), 407–414. doi:10.1016/j.gde.2004.06.014

Glasner, J. D., Liss, P., Plunkett, G. III, Darling, A., Prasad, T., & Rusch, M. (2003). ASAP, a systematic annotation package for community analysis of genomes. *Nucleic Acids Research*, *31*(1), 147–151. doi:10.1093/nar/gkg125

Glass, K., & Kauffman, S. (1973). The logical analysis of continuous, nonlinear biochemical control networks. *Journal of Theoretical Biology*, *39*, 103–129. doi:10.1016/0022-5193(73)90208-7

Glass, L. (1975). Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology*, *54*, 85–107. doi:10.1016/S0022-5193(75)80056-7

Godsill, S. J. (2001). On the relationship between Markov Chain Monte Carlo methods for model uncertainty. *Journal of Computational and Graphical Statistics*, *10*, 1–19. doi:10.1198/10618600152627924

Golding, I., Paulsson, J., Zawilski, S. M., & Cox, E. C. (2005). Real-time kinetics of gene activity in individual bacteria. *Cell*, *123*, 1025–1036. doi:10.1016/j.cell.2005.09.031

Golightly, A., & Wilkinson, D. J. (2008). Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, *52*, 1674–1693. doi:10.1016/j.csda.2007.05.019

Goncalves, A., & Costa, E. (2007). *A computational model for genetic regulatory networks* (TR 2007/06, ISSN 0874-338X). Coimbra, Portugal: Universidade de Coimbra, CISUC.

Goncalves, A., & Costa, E. (2008). A computational model of gene regulatory networks and its topological properties. In S. Bullock, J. Noble, R. Watson & M. A. Bedau (Eds.), *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems* (pp. 204-211). Cambridge, MA: MIT Press.

Gonzales, A., & Kageyema, R. (2007). Practical lessons from theoretical models about the somitogenesis. *Gene Regulation and Systems Biology*, *1*, 35–42.

Gonzalez, A., Chaouiya, C., & Thieffry, D. (2006). Dynamical analysis of the regulatory network defining the dorsal-ventral boundary of the Drosophila wing imaginal disc. *Genetics*, *174*(3), 1625–1634. doi:10.1534/genetics.106.061218

Gonzalez, A., Naldi, A., Sanchez, L., Thieffry, D., & Chaouiya, C. (2006). GINsim: A software suite for the qualitative modelling, simulation, and analysis of regulatory networks. *Bio Systems*, *84*(2), 91–100. doi:10.1016/j.biosystems.2005.10.003

Goodwin, B. C. (1965). Oscillatory behavior in enzymatic control processes. *Advances in Enzyme Regulation*, *3*, 425–438. doi:10.1016/0065-2571(65)90067-1

Goring, H. H., Curran, J. E., Johnson, M. P., Dyer, T. D., Charlesworth, J., & Cole, S. A. (2007). Discovery of expression QTLs using large-scale transcriptional profiling in human lymphocytes. *Nature Genetics*, *39*(10), 1208–1216. doi:10.1038/ng2119

Goss, P. J. E., & Peccoud, J. (1998). Quantitative modeling of stochastic systems in moldecular biology by using stochstic Petri nets. *Proceedings of the National Academy of Sciences of the United States of America*, *95*, 6750–6755. doi:10.1073/pnas.95.12.6750

Goss, P. J. E., & Peccoud, J. (1999). *Analysis of the stabilizing effect of Rom on the genetic network controlling ColE1 plasmid replication.* Paper presented at the Pacific Symposium on Biocomputing, Hawaii.

Goutsias, J., & Kim, S. (2004). A nonlinear discrete dynamical model for transcriptional regulation: Construction and properties. *Biophysical Journal*, *864*, 1922–1945. doi:10.1016/S0006-3495(04)74257-5

Goutsias, J., & Kim, S. (2006). Stochastic transcriptional regulatory systems with time delays: A mean field approximation. *Journal of Computational Biology*, *13*(5), 1049–1076. doi:10.1089/cmb.2006.13.1049

Gouze, J.-L. (1998). Positive and negative circuits in dynamical systems. *Journal of Biological System*, *6*(21), 11–15. doi:10.1142/S0218339098000054

Grafahrend-Belau, E. (2006). *Classification of T-Invariants in biochemical Petri nets based on different cluster analysis techniques (in German).* Technical University of Applied Sciences Berlin, Berlin.

Grafahrend-Belau, E., Schreiber, F., Heiner, M., Sackmann, A., Junker, B. H., & Grunwald, S. (2008). Modularization of biochemical networks based on classification of Petri net t-invariants. *Journal*, *9*, 90. doi:.doi:10.1186/1471-2105-9-90

Graham, I., & Matthai, C. C. (2003). Investigation of the forest-fire model on a small-world network. *Physical*

Review E: Statistical, Nonlinear, and Soft Matter Physics, 68, 036109. doi:10.1103/PhysRevE.68.036109

Green, P. J. (1995). Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. *Biometrika, 82*, 711–732. doi:10.1093/biomet/82.4.711

Grefenstette, J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-16*(1), 122–128. doi:10.1109/TSMC.1986.289288

Grefenstette, J., Kim, S., & Kauffman, S. (2006). An analysis of the class of gene regulatory functions implied by a biochemical model. *Bio Systems, 84*, 81–90. doi:10.1016/j.biosystems.2005.09.009

Gregory, R., Chendrimada, T., & Shiekhattar, R. (2006). MicroRNA biogenesis: Isolation and characterization of the microprocessor complex. *Methods in Molecular Biology (Clifton, N.J.), 342*, 33–47.

Grondin, Y., Raine, D. J., & Norris, V. (2007). The correlation between architecture and mRNA abundance in the genetic regulatory network of escherichia coli. *BMC Systems Biology, 1*, 30. doi:10.1186/1752-0509-1-30

Grunwald, S., Speer, A., Ackermann, J., & Koch, I. (2008). Petri net modelling of gene regulation of the Duchenne muscular dystrophy. *Bio Systems, 92*, 189–205. doi:10.1016/j.biosystems.2008.02.005

Guelzim, N., Bottani, S., Bourgine, P., & Kepes, F. (2002). Topological and causal structure of the yeast transcriptional regulatory network. *Nature Genetics, 31*(1), 60–63. doi:10.1038/ng873

Guet, C. C., Elovitz, M. B., Hsing, W., & Leibler, S. (2002). Combinatorial synthetis of genetic networks. *Science, 296*, 1466–1470. doi:10.1126/science.1067407

Guido, N. J., Wang, X., Adalsteinsson, D., McMillen, D., Hasty, J., & Cantor, C. R. (2006). A bottom-up approach to gene regulation. *Nature, 439*(16), 856–860. doi:10.1038/nature04473

Guptasarma, P. (1995). Does replication-induced transcription regulate synthesis of the myriad low copy number proteins of *Escherichia coli? BioEssays, 17*, 987–997. doi:10.1002/bies.950171112

Gustafsson, M., Hörnquist, M., & Lombardi, A. (2005). Constructing and analyzing a large-scale gene-to-gene regulatory network–lasso constrained inference and biological validation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2*(3), 254–261. doi:10.1109/TCBB.2005.35

Gustafsson, M., Hörnquist, M., Björkegren, J., & Tegnér, J. (in press). Genomewide system analysis reveals stable yet flexible network dynamics in yeast. *IET Systems Biology.*

Gustafssson, M., & Hörnquist, M. (2008). Gene expression by the elastic net. In M. Kellis, A. Califano & G. Stolovitzky (Eds.), *DREAM3, RECOMB satellite proceedings* (p. 48 and p. 133). Boston, MA.

Gustafssson, M., & Hörnquist, M. (in press). Reverse engineering of gene networks with LASSO and nonlinear basis functions. *Annals of the New York Academy of Sciences.*

Gustincich, S., Sandelin, A., Plessy, C., Katayama, S., Simone, R., & Lazarevic, D. (2006). The complexity of the mammalian transcriptome. *The Journal of Physiology, 575*(2), 321–332. doi:10.1113/jphysiol.2006.115568

Guthke, R., Möller, U., Hoffmann, M., Thies, F., & Töpfer, S. (2005). Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection. *Bioinformatics (Oxford, England), 21*(8), 1626–1634. doi:10.1093/bioinformatics/bti226

Gutierrez-Rios, R. M., Rosenblueth, D. A., Loza, J. A., Huerta, A. M., Glasner, J. D., Blattner, F. R., & Collado-Vides, J. (2003). Regulatory network of Escherichia coli: Consistency between literature knowledge and microarray profiles. *Genome Research, 13*(11), 2435–2443. doi:10.1101/gr.1387003

Haas, P. J. (2002). *Stochastic Petri nets, modelling, stability, simulation.* Berlin: Springer-Verlag.

Hack, M. H. T. (1976). *Decidability questions for Petri nets.* M.I.T.

Hallinan, J., & Wiles, J. (2004). Evolving genetic regulatory networks using an artificial genome. In Y. P. Chen (Ed.), *Proceedings of the Second Conference on Asia-Pacific Bioinformatics, 29*, 291-296. Darlinghurst, Australia: Australian Computer Society.

Hans, C., Dobra, A., & West, M. (2007). Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association, 102*(478), 507–516. doi:10.1198/016214507000000121

Harbison, C. T., Gordon, D. B., Lee, T. I., Rinaldi, N. J., Macisaac, K. D., & Danford, T. W. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, *431*(7004), 99–104. doi:10.1038/nature02800

Hardy, S., & Robillard, P. N. (2004). Modeling and simulation of molecular biology systems using Petri nets: Modeling goals of various approaches. *Journal of Bioinformatics and Computational Biology*, *2*(4), 619–637. doi:10.1142/S0219720004000764

Hart, G. T., Ramani, A. K., & Marcotte, E. M. (2006). How complete are current yeast and human protein-interaction networks? *Genome Biology*, *7*, 120. doi:10.1186/gb-2006-7-11-120

Hartemink, A. (2001). Principled search for gene regulation. Unpublished doctoral dissertation, Harvard University.

Hartemink, A. (2006). Bayesian networks and informative priors: Transcriptional regulatory network models. In K.-A. Do, P. Müller & M. Vannucci (Eds.), *Bayesian inference for gene expression and proteomics* (pp. 401-424). Cambridge: Cambridge University Press.

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2002). Combining location and expression data for principled discovery of genetic network models. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *7*, 437–449.

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing* (pp. 422-33).

Hartl, D. L., & Jones, E. W. (2005). *Essential genetics: A genomic perspective*, 4th edition. Jones & Bartlett Publishers.

Harvey, I., & Bossomaier, T. (1997). Time out of joint: Attractors in asynchronous random boolean networks. In P. Husbands & I. Harvey (Eds.), *Proceedings of the Fourth European Conference on Artificial Life* (pp. 67-75). Cambridge, MA: The MIT Press.

Hashimoto, R. F., Kim, S., Shmulevich, I., Zhang, W., Bittner, M. L., & Dougherty, E. R. (2004). Growing genetic regulatory networks from seed genes. *Bioinformatics (Oxford, England)*, *20*, 1241–1247. doi:10.1093/bioinformatics/bth074

Hassoun, M. H. (1995). *Fundamentals of artificial neural networks.* Cambridge, MA: MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction.* New York: Springer-Verlag.

Hasty, J., & Isaacs, F. (2001). Designer gene networks: Towards fundamental cellular control. *Chaos (Woodbury, N.Y.)*, *11*(1), 207–220. doi:10.1063/1.1345702

Hasty, J., McMillen, D., Isaacs, F., & Collins, J. (2001). Computational studies of gene regulatory networks: In numero molecular biology. *Nature Reviews. Genetics*, *2*(4), 268–279. doi:10.1038/35066056

Hattne, J., Fange, D., & Elf, J. (2005). Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics (Oxford, England)*, *21*, 2923. doi:10.1093/bioinformatics/bti431

Haupt, Y., Maya, R., Kazaz, A., & Oren, M. (2005). Mdm2 promotes the rapid degradation of p53. *Nature*, *387*, 296–299. doi:10.1038/387296a0

Hawley, D. K., & McClure, W. R. (1982). Mechanism of activation of transcription initiation from the λ PRM promoter. *Journal of Molecular Biology*, *157*, 493–525. doi:10.1016/0022-2836(82)90473-9

He, H., Cai, L., Skogerbø, G., Deng, W., Liu, T., & Zhu, X. (2006). Profiling Caenorhabditis elegans noncoding RNA expression with a combined microarray. *Nucleic Acids Research*, *34*(10), 2976–2983. doi:10.1093/nar/gkl371

Heckerman, D. (1999). A tutorial on learning with Bayesian networks. In M. Jordan (Ed.), *Learning in graphical models.* Cambridge: MIT Press

Heckerman, D., & Breese, J. S. (1994). Causal independence for probability assessment and inference using Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, *6*, 826–831.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197–243.

Heckerman, D., Meek, C., & Cooper, G. (2006). A Bayesian approach to causal discovery. *Innovations in Machine Learning*, 1.

Heiner, M., & Koch, I. (2004). *Petri net based model validation in systems biology.* Paper presented at the Proceedings of the 25th International Conference on Applications and Theory of Petri Nets (ICATPN 04), Bologna.

Heinrich, R., & Schuster, S. (1996). The regulation of cellular systems. New York: Chapman + Hall.

Henkel, Z. (2007). *Investigations of the finite state linear model of gene regulatory network modeling.* MS thesis, Arizona State University, Tempe, Arizona.

Hermsen, R., Tans, S., & ten Wolde, P. R. (2006). Transcriptional regulation by competing transcription factor modules. *PLoS Computational Biology*, *2*(12), 164. doi:10.1371/journal.pcbi.0020164

Herrgard, M. J., Covert, M. W., & Palsson, B. O. (2003). Reconciling gene expression data with known genome-scale regulatory network structures. *Genome Research*, *13*(11), 2423–2434. doi:10.1101/gr.1330003

Herrgård, M. J., Swainston, N., Dobson, P., Dunn, W. B., Arga, K. Y., & Arvas, M. (2008). A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nature Biotechnology*, *26*(10), 1155–1160. doi:10.1038/nbt1492

Hintze, A., & Adami, C. (2008). Evolution of complex modular biological networks. *PLoS Computational Biology*, *4*(2), e23. doi:10.1371/journal.pcbi.0040023

Hirata, H. (2002). Oscillatory expression of the bHLH factor Hes1 regulated by a negative feedback loop. *Science*, *298*(5594), 840. doi:10.1126/science.1074560

Hirata, H., Yoshiura, S., & Ohtsuka, T. (2002). Oscillatory expression of the bHLH factor Hes1 regulated by a negative feedback loop. *Science*, *298*, 840–843. doi:10.1126/science.1074560

Ho, Y. H., Constanzo, M., & Moore, L. (1999). Regulation of transcription at the Saccharomyces cerevisiae start transition by Stb1, at a Swi6-binding protein. *Molecular and Cellular Biology*, *19*, 5267–5278.

Ho, Y., Gruhler, A., Heilbut, A., & Bader, G. D. (2002). Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, *415*(6868), 180. doi:10.1038/415180a

Hofestädt, R. (1994). A Petri net application of metabolic processes. *Journal of Systems Analysis . Modelling and Simulation (Anaheim)*, *16*, 113–122.

Hofestädt, R., & Thelen, S. (1998). Quantitative modeling of biochemical networks. *In Silico Biology*, *1*(1), 39–53.

Hoffman, M. M., Khrapov, M. A., Cox, J. C., Yao, J., Tong, L., & Ellington, A. D. (2004). AANT: The amino acid-nucleotide interaction database. *Nucleic Acids Research*, *32*(Database issue), D174–D181. doi:10.1093/nar/gkh128

Hollenberg, D. (2007). On the evolution and dynamics of biological networks. *Rivista di Biologia*, *100*(1), 93–118.

Holley, S. A. (2007). The genetics and embryology of zebrafish metamerism. *Developmental Dynamics*, *236*(6), 1422–1449. doi:10.1002/dvdy.21162

Holmes, C. C., & Held, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis (Online)*, *1*, 55–67.

Holter, N. S., Maritan, A., Cieplak, M., Fedoroff, N. V., & Banavar, J. R. (2001). Dynamic modeling of gene expression data. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(4), 1693. doi:10.1073/pnas.98.4.1693

Hooper, S. D., Boue, S., Krause, R., Jensen, L. J., Mason, C. E., & Ghanim, M. (2007). Identification of tightly regulated groups of genes during Drosophila melanogaster embryogenesis. *Molecular Systems Biology*, *3*, 72. doi:10.1038/msb4100112

Horikawa, K., Ishimatsu, K., & Yoshimoto, E. (2006). Noise-resistant and synchronized oscillation of the segmentation clock. *Nature*, *441*(7094), 719–723. doi:10.1038/nature04861

Horvath, J. E., Bailey, J. A., Locke, D. P., & Eichler, E. E. (2001). Lessons from the human genome: Transitions between euchromatin and heterochromatin. *Human Molecular Genetics*, *10*(20), 2215–2223. doi:10.1093/hmg/10.20.2215

Huang, L., Guan, R. J., & Pardee, A. B. (1999). Evolution of transcriptional control from prokaryotic beginnings to eukaryotic complexities. *Critical Reviews in Eukaryotic Gene Expression*, *9*(3-4), 175–182.

Huang, S. (1999). Gene expression profiling, genetic networks and cellular states: An integrating concept for tumorigenesis and drug discovery. *Molecular Medicine (Cambridge, Mass.)*, *77*(6), 469–480.

Huang, S. (2001). Genomics, complexity and drug discovery: Insights from Boolean network models of cel-

lular regulation. *Pharmacomgenomics*, *2*(3), 203–222. doi:10.1517/14622416.2.3.203

Hubbell, E., Liu, W. M., & Mei, R. (2002). Robust estimators for expression analysis. *Bioinformatics (Oxford, England)*, *18*(12), 1585–1592. doi:10.1093/bioinformatics/18.12.1585

Huberman, B. A., Pirolli, P. L. T., Pitkow, J. E., & Luose, R. M. (1998). Strong regularities in World Wide Web surfing. *Science*, *280*, 95–97. doi:10.1126/science.280.5360.95

Hubner, N., Wallace, C. A., Zimdahl, H., Petretto, E., Schulz, H., & Maciver, F. (2005). Integrated transcriptional profiling and linkage analysis for identification of genes underlying disease. *Nature Genetics*, *37*(3), 243–253. doi:10.1038/ng1522

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., & Kitano, H. (2003). The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics (Oxford, England)*, *19*(4), 524–531. doi:10.1093/bioinformatics/btg015

Huerta, A. M., Salgado, H., Thieffry, D., & Collado-Vides, J. (1998). RegulonDB: A database on transcriptional regulation in Escherichia coli. *Nucleic Acids Research*, *26*(1), 55–59. doi:10.1093/nar/26.1.55

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., & Armour, C. D. (2000). Functional discovery via a compendium of expression profiles. *Cell*, *102*(1), 109–126. doi:10.1016/S0092-8674(00)00015-5

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics (Oxford, England)*, *19*(17), 2271–2282. doi:10.1093/bioinformatics/btg313

Hwang, D., Rust, A. G., Ramsey, S., Smith, J. J., Leslie, D. M., & Weston, A. D. (2005). A data integration methodology for systems biology. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(48), 17296–17301. doi:10.1073/pnas.0508647102

Ichinose, N., & Aihara, K. (2002). A gene network model and its design. *The 15th Workshop on Circuit and Systems* (pp. 589-593) (in Japanese).

Ideker, T. E., Thorsson, V., & Karp, R. M. (2000). Discovery of regulatory interactions through perturbation: Inference and experimental design. *Pacific Symposium on Biocomputing.*

Ideker, T., Thorsson, V., Ranish, J., Christmas, R., Buhler, J., & Eng, J. (2001). Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, *292*(5518), 929. doi:10.1126/science.292.5518.929

Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., & Barkai, N. (2002). Revealing modular organization in the yeast transcriptional network. *Nature Genetics*, *31*, 370–377.

Iliopoulos, I., Enright, A. J., & Ouzounis, C. A. (2001). Textquest: Document clustering of Medline abstracts for concept discovery in molecular biology. *Pacific Symposium on Biocomputing* (pp. 384-395).

Imoto, S., Goto, T., & Miyano, S. (2002). Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. *Pacific Symposium on Biocomputing* (pp. 175-186).

Imoto, S., Higuchi, T., Goto, T., & Miyano, S. (2006). Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. *Statistical Methodology*, *3*(1), 1–16. doi:10.1016/j.stamet.2005.09.013

Imoto, S., Higuchi, T., Goto, T., Kuhara, S., & Miyano, S. (2003). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Proceedings IEEE Computer Society Bioinformatics Conference, (CSB'03)* (pp. 104-113).

Imoto, S., Sunyong, K., Goto, T., Aburatani, S., Tashiro, K., Kuhara, S., et al. (2002). *Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network.* Paper presented at the Proc. IEEE Comput. Soc. Bioinform. Conf.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., & Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics (Oxford, England)*, *4*(2), 249–264. doi:10.1093/biostatistics/4.2.249

Ishikawa, M. (1996). Structural learning with forgetting. *Neural Networks*, *3*, 509–521. doi:10.1016/0893-6080(96)83696-3

Ito, T., Ota, K., Kubota, H., & Yamaguchi, Y. (2002). Roles for the two-hybrid system in exploration of the yeast pro-

tein interactome. *Molecular & Cellular Proteomics*, *1*(8), 561–566. doi:10.1074/mcp.R200005-MCP200

Ivanov, I., & Dougherty, E. R. (2004). Reduction mappings between probabilistic Boolean networks. *EURASIP Journal on Applied Signal Processing*, *1*, 125–131. doi:10.1155/S1110865704309182

Ivanov, I., Pal, R., & Dougherty, E. R. (2006), Applying reduction mappings in designing genomic regulatory networks. *IEEE/NLM Life Science Systems and Applications Workshop*, 1–2.

Ivanov, I., Pal, R., & Dougherty, E. R. (2007). Dynamics preserving size reduction mappings for probabilistic Boolean networks. *IEEE Transactions on Signal Processing*, *55*(5), 2310–2322. doi:10.1109/TSP.2006.890929

Iyer, V. R., Horak, C. E., Scafe, C. S., Botstein, D., Snyder, M., & Brown, P. O. (2001). Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, *409*(6819), 533–538. doi:10.1038/35054095

Jaakkola, T. S. (1997). *Variational methods for inference and estimation in graphical models.* Retrieved from ftp://ftp.ai.mit.edu/pub/users/tommi/thesis.ps.gz

Jacob, F., & Monod, J. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, *3*, 318–356.

Jaenisch, R., & Bird, A. (2003). Epigenetic regulation of gene expression: How the genome integrates intrinsic and environmental signals. *Nature Genetics*, *33*(Suppl), 245–254. doi:10.1038/ng1089

Jannink, J. L. (2005). Selective phenotyping to accurately map quantitative trait loci. *Crop Science*, *45*, 901–908. doi:10.2135/cropsci2004.0278

Jansen, R. C. (1993). Interval mapping of multiple quantitative trait loci. *Genetics*, *135*(1), 205–211.

Jansen, R. C. (2003). Studying complex biological systems using multifactorial perturbation. *Nature Reviews. Genetics*, *4*, 145–151. doi:10.1038/nrg996

Jansen, R. C., & Nap, J. P. (2001). Genetical genomics: The added value from segregation. *Trends in Genetics*, *17*(7), 388–391. doi:10.1016/S0168-9525(01)02310-1

Janssens, H., Hou, S., Jaeger, J., Kim, A. R., Myasnikova, E., Sharp, D., & Reinitz, J. (2006). Quantitative and predictive model of transcriptional control of the Drosophila melanogaster even skipped gene. *Nature Genetics*, *38*(10), 1159–1165. doi:10.1038/ng1886

Jasra, A., Stephens, D. A., & Holmes, C. C. (2007). Population-based reversible jump Markov Chain Monte Carlo. *Biometrika*, *19*, 1–21.

Jensen, F. V. (1996). *Introduction to Bayesian metworks.* New York: Springer.

Jensen, F. V., & Nielsen, T. D. (2007). *Bayesian networks and decision graphs* (2nd edition).

Jensen, M. H., Sneppen, K., & Tiana, G. (2003). Sustained oscillations and time delays in gene expression of protein Hes1. *FEBS Letters*, *541*, 176–177. doi:10.1016/S0014-5793(03)00279-5

Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., & Barabasi, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, *407*(6804), 651–654. doi:10.1038/35036627

Jia, Z., & Xu, S. (2007). Mapping quantitative trait loci for expression abundance. *Genetics*, *176*(1), 611–623. doi:10.1534/genetics.106.065599

Jiang, C., & Zeng, Z. B. (1995). Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics*, *140*(3), 1111–1127.

Jin, C., Lan, H., Attie, A. D., Churchill, G. A., Bulutuglo, D., & Yandell, B. S. (2004). Selective phenotyping for increased efficiency in genetic mapping studies. *Genetics*, *168*(4), 2285–2293. doi:10.1534/genetics.104.027524

Jin, Y., & Sendhoff, B. (2008). Evolving *in silico* bistable and oscillatory dynamics for gene regulatory network motifs. In *Proceedings of the Congress on Evolutionary Computation 2008*, Hong Kong (pp. 386-391).

Johnston, J. (1972). *Econometric methods* (2nd edition). St. Louis: McGraw-Hill.

Jong, H. D. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, *9*(1), 67–103. doi:10.1089/10665270252833208

Jordan, I. K., Marino-Ramirez, L., Wolf, Y. I., & Koonin, E. V. (2004). Conservation and coevolution in the scale-free human gene coexpression network. *Molecular Biology and Evolution*, *21*(11), 2058–2070. doi:10.1093/molbev/msh222

Jordan, M. I. (2004). Graphical models. *Statistical Science, 19,* 140–155. doi:10.1214/088342304000000026

Joyce, A. R., & Palsson, B. O. (2006). The model organism as a system: integrating Omics data sets. *Nature Reviews. Molecular Cell Biology, 7*(3), 198–210. doi:10.1038/nrm1857

Judge, G. G., & Griffiths, W. E. R.C, H., Ltkepohl, H., & Lee, T. C. (1985). *The theory and practice of econometrics.* Wiley.

Kacser, H., & Burns, J. A. (1981). The molecular basis of dominance. *Genetics, 97*(3-4), 639–666.

Kadonaga, J. T. (2004). Regulation of RNA polymerase II transcription by sequence-specific DNA binding factors. *Cell, 116*(2), 247–257. doi:10.1016/S0092-8674(03)01078-X

Kaern, M., Elston, T. C., Blake, W. J., & Collins, J. J. (2005). Stochasticity in gene expression: From theory to phenotypes. *Nature Reviews. Genetics, 6,* 451–464. doi:10.1038/nrg1615

Kalir, S. (2001). Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria. *Science, 292,* 2080–2083. doi:10.1126/science.1058758

Kandel, A. (Ed.). (1992). *Fuzzy expert systems.* CRC Press.

Kang, H. C., Chae, J. H., Lee, Y. H., Park, M. A., Shin, J. H., & Kim, S. H. (2005). Erythroid cell-specific alpha-globin gene regulation by the CP2 transcription factor family. *Molecular and Cellular Biology, 25*(14), 6005–6020. doi:10.1128/MCB.25.14.6005-6020.2005

Karatzas, I., & Shreve, S. E. (1991). *Brownian motion and stochastic calculus.* New York: Springer-Verlag.

Kass, R. E., & Wassermann, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association, 91,* 1343–1370. doi:10.2307/2291752

Kato, M., Hata, N., Banerjee, N., Futcher, B., & Zhang, M. Q. (2006). Identifying combinatorial regulation of transcription factors and binding motifs. *Genome Biology, 5,* R56. doi:10.1186/gb-2004-5-8-r56

Kato, T., Tsuda, K., & Asai, K. (2005). Selective integration of multiple biological data for supervised network inference. *Bioinformatics (Oxford, England), 21*(10), 2488–2495. doi:10.1093/bioinformatics/bti339

Katriel, I., & Bodlaender, H. L. (2006). *Online topological ordering.*

Katz, S., Irizarry, R. A., Lin, X., Tripputi, M., & Porter, M. W. (2006). A summarization approach for Affymetrix GeneChip data using a reference training set from a large, biologically diverse database. *BMC Bioinformatics, 7,* 464. doi:10.1186/1471-2105-7-464

Kauffman, S. (1969). Homeostasis and differentiation in random genetic control networks. *Nature, 224*(215), 177–178. doi:10.1038/224177a0

Kauffman, S. (1969). Metabolic stability and epigenesis in randomly constructed genetic networks. *Journal of Theoretical Biology, 22,* 437–467. doi:10.1016/0022-5193(69)90015-0

Kauffman, S. (1973). The large scale structure and dynamics of genetic control circuits: An ensemble approach. *Journal of Theoretical Biology, 44,* 167–190. doi:10.1016/S0022-5193(74)80037-8

Kauffman, S. A. (1996). *The origins of order: Self-organization and selection in evolution.* New York: Oxford University Press.

Kauffman, S. A. (2000). *Investigations.* New York: Oxford University Press.

Kauffman, S. A. (2004). A proposal for using the ensemble approach to understand genetic regulatory networks. *Journal of Theoretical Biology, 230*(4), 581–590. doi:10.1016/j.jtbi.2003.12.017

Ke, X. S., Liu, C. M., Liu, D. P., & Liang, C. C. (2003). MicroRNAs: Key participants in gene regulatory networks. *Current Opinion in Chemical Biology, 7*(4), 516–523. doi:10.1016/S1367-5931(03)00075-9

Keener, J., & Sneyd, J. (1998). *Mathematical physiology.* Springer.

Kel, A., Reymann, S., Matys, V., Nettesheim, P., Wingender, E., & Borlak, J. (2004). A novel computational approach for the prediction of networked transcription factors of aryl hydrocarbon-receptor-regulated genes. *Molecular Pharmacology, 66*(6), 1557–1572. doi:10.1124/mol.104.001677

Keller, E. F. (2005). Revisiting "scale-free" networks. *BioEssays, 27,* 1060–1068. doi:10.1002/bies.20294

Kel-Margoulis, O. V., Ivanova, T. G., Wingender, E., & Kel, A. E. (2002). Automatic annotation of genomic regulatory sequences by searching for composite clusters. *Pacific Symposium on Biocomputing*, 187-198.

Kendziorski, C. M., Chen, M., Yuan, M., Lan, H., & Attie, A. D. (2006). Statistical methods for expression quantitative trait loci (eQTL) mapping. *Biometrics*, *62*(1), 19–27. doi:10.1111/j.1541-0420.2005.00437.x

Kendziorski, C., & Wang, P. (2006). A review of statistical methods for expression quantitative trait loci mapping. *Mammalian Genome*, *17*(6), 509–517. doi:10.1007/s00335-005-0189-6

Keurentjes, J. J., Fu, J., de Vos, C. H., Lommen, A., Hall, R. D., & Bino, R. J. (2006). The genetics of plant metabolism. *Nature Genetics*, *38*(7), 842–849. doi:10.1038/ng1815

Keurentjes, J. J., Fu, J., Terpstra, I. R., Garcia, J. M., van den Ackerveken, G., & Snoek, L. B. (2007). Regulatory network construction in Arabidopsis by using genome-wide gene expression quantitative trait loci. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(5), 1708–1713. doi:10.1073/pnas.0610429104

Khan, S., Decker, K., Gillis, W., & Schmidt, C. (2003). A multiagent system-driven AI planning approach to biological pathway discovery. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling.*

Khodursky, A. B., & Bernstein, J. A. (2003). Life after transcription--revisiting the fate of messenger RNA. *Trends in Genetics*, *19*, 113–115. doi:10.1016/S0168-9525(02)00047-1

Kielbassa, J., Bortfeldt, R., Schuster, S., & Koch, I. (2008). Modeling of the U1 snRNP assembly pathway in alternative splicing in human cells using Petri nets. *Computational Biology and Chemistry.*

Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., & Tomita, M. (2003). Dynamic modeling of genetic networks using genetic algorithm and s-system. *Bioinformatics (Oxford, England)*, *19*(5), 643–650. doi:10.1093/bioinformatics/btg027

Kim, S., Dougherty, E. R., Chen, Y., Sivakumar, K., Meltzer, P., Trent, J. M., & Bittner, M. L. (2000). Multivariate measurement of gene expression relationships. *Genomics*, *67*, 201–209. doi:10.1006/geno.2000.6241

Kim, S., Li, H., Dougherty, E., Cao, N., Chen, Y., Bittner, M., & Suh, E. (2002). Can Markov chain models mimic biological regulation? *Journal of Biological System*, *10*(4), 337–357. doi:10.1142/S0218339002000676

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, *220*(4593), 671–680. doi:10.1126/science.220.4598.671

Kirschner, M., & Gerhart, J. (1998). Evolvability. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(15), 8420–8427. doi:10.1073/pnas.95.15.8420

Kishino, H., & Waddell, P. J. (2000). Correspondence analysis of genes and tissue types and finding genetic links from microarray data. In *Genome Inform Ser Workshop Genome Inform*. (pp. 83–95).

Kitano, H. (2001). *Foundations of systems biology.* Cambridge, MA: MIT Press.

Kitano, H. (2002). Computational systems biology. *Nature*, *420*(6912), 206–210. doi:10.1038/nature01254

Kitano, H. (2002). Systems biology: A brief overview. *Science*, *295*, 1662–1664. doi:10.1126/science.1069492

Kitano, H. (2007). Towards a theory of biological robustness. *Molecular Systems Biology*, *3*(137), 1–7.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, *46*, 604–632. doi:10.1145/324133.324140

Klemm, K., & Bornholdt, S., S. (2005). Topology of biological networks and reliability of information processing. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 18414. doi:10.1073/pnas.0509132102

Klingler, M., Soong, J., Butler, B., & Gergen, J. P. (1996). Disperse versus compact elements for the regulation of runt stripes in *Drosophila*. *Developmental Biology*, *177*(1), 73–84. doi:10.1006/dbio.1996.0146

Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic: Theory and applications.* Prentice Hall.

Kloosterman, W. P., & Plasterk, R. H. (2006). The diverse functions of microRNAs in animal development and disease. *Developmental Cell*, *11*(4), 441–450. doi:10.1016/j.devcel.2006.09.009

Kloster, M., Tang, C., & Wingreen, N. S. (2005). Finding regulatory modules through large-scale gene expression analysis. *Bioinformatics (Oxford, England), 21*(7), 1172–1179. doi:10.1093/bioinformatics/bti096

Knabe, J. F., Nehaniv, C. L., & Schilstra, M. J. (2008). Genetic regulatory network models of biological clocks: Evolutionary history matters. *Artificial Life*, *14*(1), 135–148. doi:10.1162/artl.2008.14.1.135

Koch, I. (2008). *Supplementary material to modeling Duchenne muscular dystrophy using Petri nets.* Retrieved from http://www.molgen.mpg.de/~koch_i/

Koch, I., & Heiner, M. (2008). Petri nets in biological network analysis. In B. Junker & F. Schreiber (Eds.), *Analysis of biological networks* (pp. 139–179). Wiley & Sons.

Koch, I., Junker, B., & Heiner, M. (2005). Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics (Oxford, England), 21*(7), 1219–1226. doi:10.1093/bioinformatics/bti145

Kohn, R., Smith, M., & Chan, D. (2001). Nonparametric regression using linear combinations of basis functions. *Statistics and Computing*, *11*, 313–322. doi:10.1023/A:1011916902934

Kornberg, R. D. (1999). Eukaryotic transcriptional control. *Trends in Cell Biology*, *9*(12), M46–M49. doi:10.1016/S0962-8924(99)01679-7

Kosik, K. S. (2006). The neuronal microRNA system. *Nature Reviews. Neuroscience*, *7*(12), 911–920. doi:10.1038/nrn2037

Kourilsky, P. (1973). Lysogenization by bacteriophage lambda: I. multiple infection and the lysogenic response. *Molecular & General Genetics*, *122*, 183–195. doi:10.1007/BF00435190

Kuhn, M., von Mering, C., Campillos, M., Jensen, L. J., & Bork, P. (2008). STITCH: Interaction networks of chemicals and proteins. *Nucleic Acids Research*, *36*(Database issue), D684. doi:10.1093/nar/gkm795

Kullander, K. (2005). Genetics moving to neuronal networks. *Trends in Neurosciences*, *28*(5), 239–247. doi:10.1016/j.tins.2005.03.001

Kulp, D., & Jagalur, M. (2006). Causal inference of regulator-target pairs by gene mapping of expression phenotypes. *BMC Genomics*, *7*(1), 125. doi:10.1186/1471-2164-7-125

Kumar, R., Reynolds, D. M., Shevchenko, A., Goldstone, S. D., & Dalton, S. (2000). Forkhead transcription factor Fkh1p and Fkh2p collaborate with Mcm1p to control transcription required for M-phase. *Current Biology, 10*, 896–906. doi:10.1016/S0960-9822(00)00618-7

Kuo, L., & Mallick, B. (1998). Variable selection for regression models. *Sankhya*, B, 60, Part 1, 65-81.

Kuo, P., & Banzhaf, W. (n.d.). Small world and scale-free network topologies in an artificial regulatory network model. In J. Pollack, M. Bedau, P. Husbands, T. Ikegami & R.A. Watson (Eds.), *Artificial Life IX: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (pp. 404-409).

Kuroe, Y., Ikeda, H., & Mori, T. (1997). Identification of nonlinear dynamical systems by recurrent high-order neural networks. In . *Proceedings of IEEE International Conference on Systems Man and Cybernetics*, *1*, 70–75.

Kurtz, T. G. (1972). The relationship between stochastic and deterministic models of chemical reactions. *The Journal of Chemical Physics*, *57*, 2976–2978. doi:10.1063/1.1678692

Kuwahara, H. (2007). *Model abstraction and temporal behavior analysis of genetic regulatory networks.* Unpublished doctoral dissertation, University of Utah.

Kuwahara, H., & Myers, C. (2007). Production-passage-time approximation: A new approximation method to accelerate the simulation process of enzymatic reactions. In *The 11th Annual International Conference on Research in Computational Molecular Biology.*

Kuwahara, H., Myers, C., & Samoilov, M. (2006). Abstracted stochastic analysis of type 1 pili expression in E. coli. In *The 2006 International Conference on Bioinformatics and Computational Biology.*

Kuwahara, H., Myers, C., Barker, N., Samoilov, M., & Arkin, A. (2005). Asynchronous abstraction methodology for genetic regulatory networks. In *The Third International Workshop on Computational Methods in Systems Biology.*

Kuwahara, H., Myers, C., Samoilov, M., Barker, N., & Arkin, A. (2006). Automated abstraction methodology for genetic regulatory networks. *Trans. on Comput. Systematic Biology, VI*, 150–175.

Kwon, Y. K., & Cho, K. H. (2008). Quantitative analysis of robustness and fragility in biological networks based on feedback dynamics. *Bioinformatics (Oxford, England), 24*(7), 987–994. doi:10.1093/bioinformatics/btn060

Lage, K., Karlberg, E. O., Størling, Z. M., Olason, P. I., Pedersen, A. G., & Rigina, O. (2007). A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature Biotechnology, 25*(3), 309–316. doi:10.1038/nbt1295

Lähdesmäki, H. (2003). On learning gene regulatory networks under the Boolean network model. *Machine Learning, 52*, 147–167. doi:10.1023/A:1023905711304

Lähdesmäki, H., Hautaniemi, S., Shmulevich, I., & Yli-Hara, O. (2006). Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Processing, 86*(4), 814–834. doi:10.1016/j.sigpro.2005.06.008

Lan, H., Stoehr, J. P., Nadler, S. T., Schueler, K. L., Yandell, B. S., & Attie, A. D. (2003). Dimension reduction for mapping mRNA abundance as quantitative traits. *Genetics, 164*(4), 1607–1614.

Lander, E. S., & Schork, N. J. (1994). Genetic dissection of complex traits. *Science, 265*(5181), 2037–2048. doi:10.1126/science.8091226

Lander, E., & Botstein, D. (1989). Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics, 121*, 185–199.

Larhlimi, A., & Bockmayr, A. (2006). *A new approach to flux coupling analysis of metabolic networks.* Paper presented at the Computational Life Sciences II, CompLife'06, Cambridge, UK.

Larraaga, P., Poza, M., Yurramendi, Y., & Murga, R. H. (1996). Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(9), 912–926. doi:10.1109/34.537345

Larsen, P., Almasri, E., Chen, G., & Dai, Y. (2007). A statistical method to incorporate biological knowledge for generating testable novel gene regulatory interactions from microarray experiments. *BMC Bioinformatics, 8*, 317. doi:10.1186/1471-2105-8-317

Lavine, M. L. (1999). What is Bayesian statistics and why everything else is wrong. *Journal of Undergraduate Mathematics and Its Applications, 20*, 165–174.

Le, P. P., Bahl, A., & Unga, L. H. (2004). Using prior knowledge to improve genetic network reconstruction from microarray data. *In Silico Biology, 4*, 0027.

Lee, D.-S., & Rieger, H. (2007). Comparative study of the transcriptional regulatory networks of E. coli and yeast: Structural characteristics leading to marginal dynamic stability. *Journal of Theoretical Biology, 248*(4), 618–626. doi:10.1016/j.jtbi.2007.07.001

Lee, D.-Y., Zimmer, R., Lee, D.-Y., Hanisch, D., & Sunwon, P. (2004). Knowledge representation model for systems-level analysis of signal transduction networks. *Genome Informatics, 15*(2), 234–243.

Lee, H. K., Hsu, A. K., Sajdak, J., Qin, J., & Pavlidis, P. (2004). Coexpression analysis of human genes across many microarray data sets. *Genome Research, 14*(6), 1085–1094. doi:10.1101/gr.1910904

Lee, K. E., Sha, N., Dougherty, E. R., Vannucci, M., & Mallick, B. K. (2003). Gene selection: A Bayesian variable selection approach. *Bioinformatics (Oxford, England), 19*(1), 90–97. doi:10.1093/bioinformatics/19.1.90

Lee, S. I., Pe'er, D., Dudley, A. M., Church, G. M., & Koller, D. (2006). Identifying regulatory mechanisms using individual variation reveals key role for chromatin modification. *Proceedings of the National Academy of Sciences of the United States of America, 103*(38), 14062–14067. doi:10.1073/pnas.0601852103

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., & Gerber, G. K. (2002). Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science, 298*(5594), 799–804. doi:10.1126/science.1075090

Leier, A., Márquez-Lago, T., & Burrage, K. (2008). Generalized binomial τ-leap method for biochemical kinetics incorporating both delay and intrinsic noise. *The Journal of Chemical Physics, 128*, 205107. doi:10.1063/1.2919124

Leier, A., Márquez-Lago, T., & Burrage, K. (2008). Modeling intrinsic noise and delays in chemical kinetics of coupled autoregulated oscillating cells. *Int. J. Multiscale Computational Engineering, 6*(1).

Levine, M., & Tjian, R. (2003). Transcription regulation and animal diversity. *Nature, 424*, 147–151. doi:10.1038/nature01763

Levsky, J. M., Shenoy, S. M., Pezo, R. C., & Singer, R. H. (2002). Single-cell gene expression profiling. *Science, 297*, 836–840. doi:10.1126/science.1072241

Lewis, J. (2003). Autoinhibition with transcriptional delay: A simple mechanism for the zebrafish somitogenesis oscillator. *Current Biology*, *13*, 1398–1408. doi:10.1016/S0960-9822(03)00534-7

Lewis, J., & Ozbudak, E. M. (2007). Deciphering the somite segmentation clock: Beyond mutants and morphants. *Developmental Dynamics*, *236*(6), 1410–1415. doi:10.1002/dvdy.21154

Li, F., Long, T., Lu, Y., Ouyang, Q., & Tang, C. (2004). The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(14), 4781–4786. doi:10.1073/pnas.0305937101

Li, H., Chen, H., Bao, L., Manly, K. F., Chesler, E. J., & Lu, L. (2006). Integrative genetic analysis of transcription modules: towards filling the gap between genetic loci and inherited traits. *Human Molecular Genetics*, *15*(3), 481–492. doi:10.1093/hmg/ddi462

Li, H., Lu, L., Manly, K. F., Chesler, E. J., Bao, L., & Wang, J. (2005). Inferring gene transcriptional modulatory relations: A genetical genomics approach. *Human Molecular Genetics*, *14*(9), 1119–1125. doi:10.1093/hmg/ddi124

Li, P., Zhang, C., Perkins, E. J., Gong, P., & Deng, P. (2007). Comparison of probabilistic Boolean network and dynamic Bayesian network approaches for inferring gene regulatory networks. *BMC Bioinformatics*, *8*(Suppl 7), S13. doi:10.1186/1471-2105-8-S7-S13

Li, W., & Nyholt, D. R. (2001). Marker selection by Akaike information criterion and Bayesian information criterion. *Genetic Epidemiology*, *21*(Suppl. 1), S272–S277.

Li, X. Y., MacArthur, S., Bourgon, R., Nix, D., Pollard, D. A., & Iyer, V. N. (2008). Transcription factors bind thousands of active and inactive regions in the Drosophila blastoderm. *PLoS Biology*, *6*(2), e27. doi:10.1371/journal.pbio.0060027

Li, Y., Alvarez, O. A., Gutteling, E. W., Tijsterman, M., Fu, J., & Riksen, J. A. (2006b). Mapping determinants of gene expression plasticity by genetical genomics in C. elegans. *PLOS Genetics*, *2*(12), e222. doi:10.1371/journal.pgen.0020222

Li, Y., Campbell, C., & Tipping, M. (2002). Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics (Oxford, England)*, *18*(10), 1332–1339. doi:10.1093/bioinformatics/18.10.1332

Liang, S., Fuhrman, S., & Somogyi, R. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Proc. Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *3*, 18–29.

Lieb, J. D., Liu, X., Botstein, D., & Brown, P. O. (2001). Promoter-specific binding of Rap1 revealed by genome-wide maps of protein-DNA association. *Nature Genetics*, *28*(4), 327–334. doi:10.1038/ng569

Liebovitch, L. S., Jirsa, V. K., & Shehadeh, L. A. (2006). Structure of genetic regulatory networks: Evidence for scale free networks. In M. M. Nowak (Ed.), *Complexus mundi: Emergent patterns in nature* (pp. 1-8). Singapore: World Scientific.

Liebovitch, L. S., Tsinoremas, N., & Pandya, A. (2007). Developing combinatorial multicomponent therapies (CMCT) of drugs that are more specific and have fewer side effects than traditional one drug therapies. *Nonlinear Biomedical Physics*, *1*, 11. doi:. doi:10.1186/1753-4631-1-11

Lipshtat, A., Loinger, A., Balaban, N. Q., & Biham, O. (2006). Genetic toggle switch without cooperative binding. *Physical Review Letters*, *96*, 188101. doi:10.1103/PhysRevLett.96.188101

Little, J. W., Shepley, D. P., & Wert, D. W. (1999). Robustness of a gene regulatory circuit. *The EMBO Journal*, *18*, 4299–4307. doi:10.1093/emboj/18.15.4299

Liu, B., de la Fuente, A., & Hoeschele, I. (2008). Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics*, *178*, 1763–1776. doi:10.1534/genetics.107.080069

Liu, K.-Y., Zhou, X., Kan, K., & Wong, S. T. C. (2006). Bayesian variable selection for gene expression modeling with regulatory motif binding sites in neuroinflammatory events. *Neuroinformatics*, *6*, 95–117. doi:10.1385/NI:4:1:95

Liu, T., Lin, Y., Wen, X., Jorissen, R. N., & Gilson, M. K. (2007). BindingDB: A Web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Research*, *35*(Suppl 1), D198–D201. doi:10.1093/nar/gkl999

Lodish, H., Berk, A., Zipursky, L. S., Matsudaira, P., Baltimore, D., & Darnell, J. (1999). *Molecular cell biology*. W. H. Freeman and Company.

Logan, N., Delavaine, L., Graham, A., Reilly, C., Wilson, J., & Brummelkamp, T. R. (2004). E2F-7: A distinctive E2F family member with an unusual organization of DNA-binding domains. *Oncogene, 23*(30), 5138–5150. doi:10.1038/sj.onc.1207649

Lu, J., Engl, H. W., & Schuster, P. (2006). Inverse bifurcation analysis: Application to simple gene systems. *Alg. Mol. Biol., 1*(11).

Lu, J., Ruhf, M. L., Perrimon, N., & Leder, P. (2007). A genome-wide RNA interference screen identifies putative chromatin regulators essential for E2F repression. *The Proceedings of the National Academy of Sciences Online (US), 104*(22), 9381–9386. doi:10.1073/pnas.0610279104

Lucas, J., Carvalho, C., Wang, Q., Bild, A., & West, M. (2006). Sparse statistical modeling in gene expression genomics. In P. M. a. M. V. e. KA Do (Eds.), *Bayesian inference for gene expression and proteomics*. Cambridge: Cambridge University Press.

Lum, P. Y., Chen, Y., Zhu, J., Lamb, J., Melmed, S., & Wang, S. (2006). Elucidating the murine brain transcriptional network in a segregating mouse population to identify core functional modules for obesity and diabetes. *Journal of Neurochemistry, 97*(Suppl 1), 50–62. doi:10.1111/j.1471-4159.2006.03661.x

Lundström, J. (2007). *Private communication.*

Luo, Z. W., Potokina, E., Druka, A., Wise, R., Waugh, R., & Kearsey, M. J. (2007). SFP genotyping from affymetrix arrays is robust but largely detects cis-acting expression regulators. *Genetics, 176*(2), 789–800. doi:10.1534/genetics.106.067843

Luscombe, N. M., Babu, M. M., Yu, H., Snyder, M., Teichmann, S. A., & Gerstein, M. (2004). Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature, 431*(7006), 308–312. doi:10.1038/nature02782

Luscombe, N., Laskowski, R., & Thornton, J. (2001). Amino acid-base interactions: A three-dimensional analysis of protein-DNA interactions at an atomic level. *Nucleic Acids Research, 29*, 2860–2874. doi:10.1093/nar/29.13.2860

Lynch, A. S., & Lin, E. C. C. (1996). Responses to molecular oxygen. In *Escherichia coli and salmonella: Cellular and molecular biology* (pp. 1526–1538). Washington, D. C.: American Society for Microbiology, 2nd edition.

Lynch, M. (2007). *The origins of genome architecture*. Sunderland, MA: Sinaur Associates Inc.

Ma, H. W., & Zeng, A. P. (2003). The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics (Oxford, England), 19*(11), 1423–1430. doi:10.1093/bioinformatics/btg177

Ma, H.-W., Buer, J., & Zeng, A.-P. (2004). Hierarchical structure and modules in the escherichia coli transcriptional regulatory network revealed by a new top-down approach. *BMC Bioinformatics, 5*, 199. doi:10.1186/1471-2105-5-199

Ma, L., Wagner, J., Rice, J., Hu, W., Levine, J., & Stolovitzky, G. (2005). A plausible model for the digital response of p53 to DNA damage. *Proceedings of the National Academy of Sciences of the United States of America, 102*(4), 14266–14271. doi:10.1073/pnas.0501352102

Ma'ayan, A., Lipshtat, A., Iyengar, R., & Sontag, E. D. (2008). Proximity of intracellular regulatory networks to monotone systems. *IET Systems Biology, 2*, 103–112. doi:10.1049/iet-syb:20070036

MacNamara, S., Burrage, K., & Sidje, R. (2008). Multiscale modeling of chemical kinetics via the master equation. *SIAM J. Multiscale Modelling and Simulation Multiscale Modeling & Simulation, 6*(4).

Madan Babu, M., & Teichmann, S. A. (2003). Functional determinants of transcription factors in Escherichia Coli: Protein families and binding sites. *Trends in Genetics, 19*(2), 75–79. doi:10.1016/S0168-9525(02)00039-2

Madigan, D., & Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association, 89*(428), 1535–1546. doi:10.2307/2291017

Mahler, M., Most, C., Schmidtke, S., Sundberg, J. P., Li, R., & Hedrich, H. J. (2002). Genetics of colitis susceptibility in IL-10-deficient mice: Backcross versus F2 results contrasted by principal component analysis. *Genomics, 80*(3), 274–282. doi:10.1006/geno.2002.6840

Mancosu, G., Pieroni, E., Maggio, F., Fotia, G., Liu, B., Hoeschele, I., et al. (2008). Deciphering a genome-wide yeast gene network. *Submitted.*

Mangin, B., Thoquet, P., & Grimsley, N. H. (1998). Pleiotropic QTL analysis. *Biometrics*, *54*, 88–99. doi:10.2307/2533998

Manke, T., Bringos, R., & Virigron, M. (2005). Correlating protein-DNA and protein-protein interactions networks. *Journal of Molecular Biology, 333*, 7585.

Margaliot, M. (2007). Mathematical modeling of natural phenomena: A fuzzy logic approach. In P. P. Wang, D. Ruan & E. E. Kerre (Eds.), *Fuzzy logic-a spectrum of theoretical and practical issues* (pp. 113-134). Springer.

Margaliot, M. (2008). Biomimicry and fuzzy modeling: A match made in heaven. *IEEE Computational Intelligence Magazine*, *3*, 38–48. doi:10.1109/MCI.2008.926602

Margaliot, M., & Langholz, G. (1999). Fuzzy Lyapunov based approach to the design of fuzzy controllers. *Fuzzy Sets and Systems*, *106*, 49–59. doi:10.1016/S0165-0114(98)00356-X

Margaliot, M., & Langholz, G. (2000). *New approaches to fuzzy modeling and control-design and analysis.* World Scientific.

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., & Dalla Favera, R. (2006). ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics, 7*(Suppl 1), S7. doi:10.1186/1471-2105-7-S1-S7

Margolin, A. A., Palomero, T., Ferrando, A. A., Califano, A., & Stolovitzky, G. (2007). ChIP-on-chip significance analysis reveals ubiquitous transcription factor binding. *BMC Bioinformatics, 8*(Suppl 8), S2. doi:10.1186/1471-2105-8-S8-S2

Marguerat, S., Jensen, T. S., de Lichtenberg, U., Wilhelm, B. T., Jensen, L. J., & Bähler, J. (2006). The more the merrier: Comparative analysis of microarray studies on cell cycle-regulated genes in fission yeast. *Yeast (Chichester, England), 23*, 261–277. doi:10.1002/yea.1351

Maritz, J. S., & Lwin, T. (1989). *Empirical Bayes methods.* Chapman & Hall, 2nd edition.

Markstein, M., Markstein, P., Markstein, V., & Levine, M. (2002). Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the Drosophila embryo. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(2), 763–768. doi:10.1073/pnas.012591199

Maroulakou, I. G., & Bowe, D. B. (2000). Expression and function of Ets transcription factors in mammalian development: A regulatory network. *Oncogene, 19*(55), 6432–6442. doi:10.1038/sj.onc.1204039

Marquez-Lago, T., & Burrage, K. (2007). Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics. *The Journal of Chemical Physics, 127*(10), 104101. doi:10.1063/1.2771548

Marr, C., & Hütt, M.-Th. (2005). Topology regulates pattern formation capacity of binary cellular automata on graphs. *Physica A, 354*, 641–662. doi:10.1016/j.physa.2005.02.019

Marr, C., & Hütt, M.-Th. (2006). Similar impact of topological and dynamic noise on complex patterns. *Physics Letters. [Part A], 349*, 302–305. doi:10.1016/j.physleta.2005.08.096

Marr, C., Geertz, M., Hütt, M.-T., & Mushkhelishvili. (2008). Dissecting the logical types of network control in gene expression profiles. *BMC Sys. Biol.*, 2, 18.

Martelli, A. M., Nyåkern, M., Tabellini, G., Bortul, R., Tazzari, P. L., Evangelisti, C., & Cocco, L. (2006). Phosphoinositide 3-kinase/Akt signaling pathway and its therapeutical implications for human acute myeloid leukemia. *Leukemia*, *20*(6), 911–928. doi:10.1038/sj.leu.2404245

Martens, J., Laprade, L., & Winston, F. (2004). Intergenic transcription is required to repress the Saccharomyces cerevisiae SER3 gene. *Nature*, *429*, 571–574. doi:10.1038/nature02538

Martin, S., Zhang, Z., Martino, A., & Faulon, J. L. (2007). Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics (Oxford, England), 23*(7), 866–874. doi:10.1093/bioinformatics/btm021

Marwan, W., Sujathab, A., & Starostzik, C. (2005). Reconstructing the regulatory network controlling commitment and sporulation in Physarum polycephalum based on hierarchical Petri net modeling and simulation. *Journal of Theoretical Biology, 236*, 349–365. doi:10.1016/j.jtbi.2005.03.018

Masamizu, Y., Ohtsuka, T., & Takashima, Y. (2006). Real-time imaging of the somite segmentation clock: Revelation of unstable oscillators in the individual presomitic mesoderm cells. *Proceedings of the National Academy of*

Sciences of the United States of America, *103*(5), 1313–1318. doi:10.1073/pnas.0508658103

Masoud, N., Zadeh, L., & Korotkikh, V. (Eds.). (2004). *Fuzzy partial differential equations and relational equations.* Springer.

Mata, J., Marguerat, S., & Bahler, J. (2005). Post-transcriptional control of gene expression: A genomewide perspective. *Trends in Biochemical Sciences*, *30*, 506–514. doi:10.1016/j.tibs.2005.07.005

Matsuno, H., Doi, A., Nagasaki, M., & Miyano, S. (2000). *Hybrid Petri net representation of gene regulatory network*. Paper presented at the Pacific Symposium on Biocomputing, Hawaii.

Matsuno, H., Li, C., & Miyano, S. (2006). Petri net based descriptions for systematic understanding of biological pathways. *IEICE Transactions on Fundamentals of Electronics, Communication, and Computer Sciences . E (Norwalk, Conn.), 89-A*(11), 3166–3174.

Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., & Miyano, S. (2003). Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biology, 3*(3), 389–404.

Mattick, J. (2003). Introns and noncoding RNAs: The hidden layer of Eukariotic complexity. In J. Barciszewski (Ed.), *Noncoding RNAs.* Kluwer Academic.

Matys, V., Fricke, E., & Geffers, R. (2003). TRANSFAC: Transcriptional regulation, from patterns to profiles. *Nucleic Acids Research*, *31*, 374–378. doi:10.1093/nar/gkg108

Matys, V., Kel-Margoulis, O. V., Fricke, E., Liebich, I., Land, S., & Barre-Dirrie, A. (2006). TRANSFAC and its module TRANSCompel: Transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, *34*(Database issue), D108–D110. doi:10.1093/nar/gkj143

Maxam, A. M., & Gilbert, W. (1977). A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America*, *74*, 560–564. doi:10.1073/pnas.74.2.560

McAdams, H. H., & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(3), 814–819. doi:10.1073/pnas.94.3.814

McAdams, H. H., & Shapiro, L. (1995). Circuit simulation of genetic networks. *Science*, *269*, 650–656. doi:10.1126/science.7624793

McAdams, H., & Arkin, A. (1997). Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *94*, 814–819. doi:10.1073/pnas.94.3.814

McCullagh, P., & Nelder, J. A. (1983). *Generalized linear models.* London: Chapman & Hall.

McCulloch, G. E. R. (1996). Stochastic search variable selection. In S. R. a. D. S. WR Gilks (Ed.), *Markov chain Monte Carlo in practice* (pp. pp. 203-214). Boca Raton, FL: Chapman & Hall.

mCho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., & Wodicka, L. (1998). A genomewide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, *2*(1), 65–73. doi:10.1016/S1097-2765(00)80114-8

McMillan, K. L. (1993). *Symbolic model checking.* Kluwer.

McMillan, K. L. (1999). *The SMV system*. Cadence Berkeley Labs.

McMillan, K. L. (2003). Interpolation and SAT-based model checking. In CAV (Vol. 2725 of LNCS, pp. 1–13). Springer.

McQuarrie, D. A. (1967). Stochastic approach to chemical kinetics. *Journal of Applied Probability*, *4*, 413–478. doi:10.2307/3212214

Medugorac, I., & Soller, M. (2001). Selective genotyping with a main trait and a correlated trait. *Journal of Animal Breeding and Genetics*, *118*(5), 285–295. doi:10.1046/j.1439-0388.2001.00308.x

Meek, C., & Heckerman, D. (1997). Structure and parameter learning for causal independence and causal interaction models. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence* (pp. 366-375).

Mehrabian, M., Allayee, H., Stockton, J., Lum, P. Y., Drake, T. A., & Castellani, L. W. (2005). Integrating genotypic and expression data in a segregating mouse population to identify 5-lipoxygenase as a susceptibility gene for obesity and bone traits. *Nature Genetics*, *37*(11), 1224–1233. doi:10.1038/ng1619

Meinel, C., & Theobald, T. (1998). *Algorithms and data structures in VLSI design: OBDD–foundations and applications.* Berlin, Heidelberg, New York: Springer-Verlag.

Mellor, J. (2006). Dynamic nucleosomes and gene transcription. *Trends in Genetics*, *22*(6), 320–329. doi:10.1016/j.tig.2006.03.008

Menasche, G., Feldmann, J., Houdusse, A., Desaymard, C., Fischer, A., Goud, B., & de Saint Basile, G. (2003). Biochemical and functional characterization of Rab27a mutations occurring in Griscelli syndrome patients. *Blood*, *101*(7), 2736–2742. doi:10.1182/blood-2002-09-2789

Mesot, B., & Teuscher, C. (2003). Critical values in asynchronous random boolean networks. In W. Banzhaf (Ed.), *Advances in Artificial Life, ECAL2003.* (LNAI, pp. 367-376). Berlin: Springer.

Mestl, T., Plahte, E., & Omholt, S. W. (1995). A mathematical framework for describing and analyzing gene regulatory networks. *Journal of Theoretical Biology*, *176*, 291–300. doi:10.1006/jtbi.1995.0199

Michaelis, L., & Menten, M. (1913). Die Kinetik der Invertinwirkung. *Biochemische Zeitschrift*, *49*, 333–369.

Miles, M. C., Janket, M. L., Wheeler, E. D., Chattopadhyay, A., Majumder, B., & Dericco, J. (2005). Molecular and functional characterization of a novel splice variant of ANKHD1 that lacks the KH domain and its role in cell survival and apoptosis. *The FEBS Journal*, *27*(16), 4091–4102. doi:10.1111/j.1742-4658.2005.04821.x

Milgram, S. (1967, May). The small world problem. *Psychology Today*, 60–67.

Miller, J. F. (2003). Evolving developmental programs for adaptation, morphogenesis, and self-repair. In . *Proceedings of the European Conference on Artificial Life, ECAL*, *2003*, 256–265.

Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., & Ayzenshtat, I. (2004). Superfamilies of evolved and designed networks. *Science*, *303*, 1538. doi:10.1126/science.1089167

Milo, R., Shen-Orr, S., Itzkovitz, N., Kashtan, D., Chklovskii, D., & Alon, U. (2002). Network motifs: Simple building blocks of complex networks. *Science*, *298*, 824–827. doi:10.1126/science.298.5594.824

Mitchell, M., Crutchfield, J. P., & Hraber, P. T. (1994). Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D. Nonlinear Phenomena*, *75*, 361–391. doi:10.1016/0167-2789(94)90293-3

Mnaimneh, S., Davierwala, A. P., Haynes, J., Moffat, J., Peng, W. T., & Zhang, W. (2004). Exploration of essential gene functions via titratable promoter alleles. *Cell*, *118*(1), 31–44. doi:10.1016/j.cell.2004.06.013

Mo, M. L., & Palsson, B. O. (2008). Understanding human metabolic physiology: a genome-to-systems approach. *Trends Biotechnology*. Retrieved from http://dx.doi.org/10.1016/j.tibtech.2008.09.007

Monk, N. (2003). Oscillatory expression of Hes1, p53, and NF-κB driven by transcriptional time delays. *Current Biology*, *13*, 1409–1413. doi:10.1016/S0960-9822(03)00494-9

Montiel, G., Gantet, P., Jay-Allemand, C., & Breton, C. (2004). Transcription factor networks. Pathways to the knowledge of root development. *Plant Physiology*, *136*(3), 3478–3485. doi:10.1104/pp.104.051029

Moreira, A. A., Mathur, A., Diermeier, D., & Amaral, L. A. N. (2004). Efficient system-wide coordination in noisy environments. *Proceedings of the National Academy of Sciences of the United States of America*, *101*, 12085. doi:10.1073/pnas.0400672101

Moreno, Y., Pastor-Satorras, R., & Vespignani, A. (2002). Epidemic outbreaks in complex heterogeneous networks. *The European Physical Journal B*, *26*, 521.

Morgan, X. C., Ni, S., Miranker, D. P., & Iyer, V. R. (2007). Predicting combinatorial binding of transcription factors to regulatory elements in the human genome by association rule mining. *BMC Bioinformatics*, *8*, 445. doi:10.1186/1471-2105-8-445

Mori, Y., Kuroe, Y., & Mori, T. (2006). A synthesis method of gene networks based on gene expression by network learning. In *Proceedings of SICE-ICASE International Joint Conference* (pp. 4545–4550).

Morse, D. E., Mosteller, R. D., & Yanofsky, C. (1969). Dynamics of synthesis, translation, and degradation of trp operon messenger RNA in E. coli. *Cold Spring Harbor Symposia on Quantitative Biology*, *34*, 725–740.

Morton-Firth, C. J., & Bray, D. (1998). Predicting temporal fluctuations in an intracellular signalling pathway. *Journal of Theoretical Biology*, *192*, 117–128. doi:10.1006/jtbi.1997.0651

Müller-Linow, M., Hilgetag, C., & Hütt, M.-Th. (2008). (in press). Organization of excitable dynamics in hierarchical biological networks. *PLoS Computational Biology*.

Müller-Linow, M., Marr, C., & Hütt, M.-Th. (2006). Topology regulates synchronization patterns in excitable dynamics on graphs. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *74*, 016112. doi:10.1103/PhysRevE.74.016112

Mukherjee, S., & Speed, T. P. (2008). Network inference using informative priors. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(38), 14313–14318. doi:10.1073/pnas.0802272105

Muller, C. W. (2001). Transcription factors: Global and detailed views. *Current Opinion in Structural Biology*, *11*(1), 26–32. doi:10.1016/S0959-440X(00)00163-9

Muller, F., & Tora, L. (2004). The multicoloured world of promoter recognition complexes. *The EMBO Journal*, *23*(1), 2–8. doi:10.1038/sj.emboj.7600027

Munsky, B., & Khammash, M. (2006). The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics*, *124*, 044104. doi:10.1063/1.2145882

Murata, T. (1989). *Petri Nets: Properties, analysis, and applications.* Paper presented at the IEEE-International Conference on Consumer Electronics. Digest of Technical Papers.

Murphy, K. P. (2002). *Dynamic Bayesian networks*: *Representation, inference, and learning.* Unpublished doctoral thesis, UC Berkeley.

Murphy, K., & Mian, S. (1999). *Modelling gene expression data using dynamic Bayesian networks* (Tech. Rep.). Computer Science Division, University of California, Berkeley, CA.

Myers, C. J., Belluomini, W., Killpack, K., Mercer, E., Peskin, E., & Zheng, H. (2001). Timed circuits: A new paradigm for high-speed design.

Nachman, I., Regev, A., & Friedman, N. (2004). Inferring quantitative models of regulatory networks from expression data. *Bioinformatics (Oxford, England)*, *20*(Suppl 1), i248–i256. doi:10.1093/bioinformatics/bth941

Nadeau, J. H., Burrage, L. C., Restivo, J., Pao, Y. H., Churchill, G., & Hoit, B. D. (2003). Pleiotropy, homeostasis, and functional networks based on assays of cardiovascular traits in genetically randomized populations. *Genome Research*, *13*(9), 2082–2091. doi:10.1101/gr.1186603

Nagasaki, M., Atsushi Doi, A., Matsuno, H., & Miyano, S. (2003). Genomic object net I: A platform for modeling and simulating biopathways. *Applied Bioinformatics*, *2*(3), 181–184.

Nakayama, H., Tanaka, H., & Ushio, T. (2006). The formulation of the control of an expression pattern in a gene network by propositional calculus. *Journal of Theoretical Biology*, *240*, 443–450. doi:10.1016/j.jtbi.2005.10.014

Nariai, N., Kim, S., Imoto, S., & Miyano, S. (2004). Using protein-protein interactions for refining gene networks estimated from microarray data by Bayesian networks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 336–347.

Nariai, N., Tamada, Y., Imoto, S., & Miyano, S. (2005). Estimating gene regulatory networks and protein-protein interactions of Saccharomyces cerevisiae from multiple genome-wide data. *Bioinformatics (Oxford, England)*, *21*(Suppl 2), ii206–ii212. doi:10.1093/bioinformatics/bti1133

Neale, M. C., Boker, S. M., Xie, G., & Maes, H. H. (2003). *Mx: Statistical modeling.* Richmond, VA: Department of Psychiatry.

Neapolitan, R. E. (2003). *Learning Bayesian networks.* Prentice Hall.

Newman, J. R. S., Ghaemmaghami, S., Ihmels, J., Breslow, D. K., Noble, M., DeRisi, J. L., & Weissman, J. S. (2006). Single-cell proteomic analysis of s. cerevisiae reveals the architecture of biological noise. *Nature*, *441*(7095), 840–846. doi:10.1038/nature04785

Newman, M. (2003). The structure and function of complex networks. *SIAM Review*, *45*(2), 167–256. doi:10.1137/S003614450342480

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(23), 8577–8582. doi:10.1073/pnas.0601602103

Newman, M., Strogatz, S., & Watts, D. (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *64*, 026118. doi:10.1103/PhysRevE.64.026118

Nguyen, N., Kuwahara, H., Myers, C., & Keener, J. (2007). The design of a genetic muller C-element. In *The 13th IEEE International Symposium on Asynchronous Circuits and Systems.*

Nicolau, D. V. Jr, Burrage, K., & Parton, R. G. (2006). Identifying optimal lipid raft characteristics required to promote nanoscale protein-protein interactions on the plasma membrane. *Molecular and Cellular Biology*, *26*, 313–323. doi:10.1128/MCB.26.1.313-323.2006

Nicolis, G. (1995). *Introduction to nonlinear science*. Cambridge University Press.

Nilsson, N. (1980). *Principles of artificial intelligence*. Morgan Kaufmann.

Nishikawa, T., & Motter, A. E. (2006). Synchronization is optimal in nondiagonalizable networks . *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *73*, 065106. doi:10.1103/PhysRevE.73.065106

Noble, D. (2006). The music of life. Oxford.

Noman, N., & Iba, H. (2005). Inference of gene regulatory networks using s-system and differential evolution. In H. Beyer (Ed.), *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation* (pp. 439-446). USA: ACM.

Novak, V. (2005). Are fuzzy sets a reasonable tool for modeling vague phenomena? *Fuzzy Sets and Systems*, *156*, 341–348. doi:10.1016/j.fss.2005.05.029

Novikov, E., & Barillot, E. (2008). Regulatory network reconstruction using an integral additive model with flexible kernel functions. *BMC Systems Biology*, *2*(8).

Ntzoufras, I. (1999). *Gibbs variable selection using BUGS* (Tech. Rep.). Retrieved from http://www.ba.aegean.gr/ntzoufras/tr.htm

Nunoshiba, T., Hidalgo, E., Li, Z., & Demple, B. (1993). Negative autoregulation by the escherichia coli soxs protein: A dampening mechanism for the soxrs redox stress response. *Journal of Bacteriology*, *175*(22), 7492–7494.

Nutsch, T., Oesterhelt, D., Gilles, E. D., & Marwan, W. (2005). A quantitative model of the switch cycle of an archaeal flagellar motor and its sensory control. *Biophysical Journal*, *89*, 2307–2323. doi:10.1529/biophysj.104.057570

Oliveira, J. S., Bailey, C. G., Jones-Oliveira, J. B., Dixon, D. A., Gull, D. W., & Chandel, M. L. (2003). A computational model for the identification of biochemical pathways in the Krebs cycle. *Journal of Computational Biology*, *10*(1), 57–82. doi:10.1089/106652703763255679

Olivieri, P., & Davidson, E. H. (2004). Genes regulatory network controling embryonic specification in the sea urchin. *Current Opinion in Genetics & Development*, *14*, 351–360. doi:10.1016/j.gde.2004.06.004

Olson, E. N. (2006). Gene regulatory networks in the evolution and development of the heart. *Science*, *313*(5795), 1922–1927. doi:10.1126/science.1132292

Omholt, S. W., Plahte, E., Oyehaug, L., & Xiang, K. (2000). Gene regulatory networks generating the phenomena of additivity, dominance and epistasis. *Genetics*, *155*(2), 969–980.

Onami, S., Kyoda, K. M., Morohashi, M., & Kitano, H. (2001). The DBRF method for inferring a gene network from large-scale steady-state gene expression data. In H. Kitano (Ed.), *Foundations of systems biology* (pp. 59-75). Cambridge, MA.: The MIT Press.

Oosawa, C., & Savageau, M. A. (2002). Effects of alternative connectivity on behavior of randomly constructed Boolean networks. *Physica D. Nonlinear Phenomena*, *170*, 143–161. doi:10.1016/S0167-2789(02)00530-4

Ota, K., Yamada, T., Yamanishi, Y., Goto, S., & Kanehisa, M. (2003). Comprehensive analysis of delay in transcriptional regulation using expression profiles. *Genome Inform.*, *14*, 302–303.

Ozbudak, E. M., Thattai, M., Kurtser, I., Grossman, A. D., & Oudenaarden, A. v. (2002). Regulation of noise in gene expression of a single gene. *Nature Genetics*, *31*, 69–73. doi:10.1038/ng869

Pal, R., Data, A., Fornace, A. J., Bittner, M. L., & Dougherty, E. R. (2005). Boolean relationships among genes responsive to ionizing radiation in the NCI 60 ACDS. *Bioinformatics (Oxford, England)*, *21*(8), 1542–1549. doi:10.1093/bioinformatics/bti214

Pal, R., Ivanov, I., & Dougherty, E. R. (2005). Generating Boolean networks with a prescribed attractor structure. *Bioinformatics (Oxford, England)*, *21*(21), 4021–4025. doi:10.1093/bioinformatics/bti664

Paladugu, S. R., Chickarmana, V., Deckard, A., Frumkin, J. P., McCormack, M., & Sauro, H. M. (2006). *In silico* evolution of functional modules in biological networks. *IEE Proc.- . Systematic Biology, 153*(4), 223–235.

Parikh, R. J. (1966). On context-free languages. *Journal of the association for computing machinery, 13*, 570–581.

Parker, R., & Song, H. (2004). The enzymes and control of eukaryotic mRNA turnover. *Nature Structural & Molecular Biology, 11*, 121–127. doi:10.1038/nsmb724

Parkhomenko, E., Tritchler, D., & Beyene, J. (2007). Genome-wide sparse canonical correlation of gene expression with genotypes. *BMC Proceedings, 1*(Suppl 1), S119. doi:10.1186/1753-6561-1-s1-s119

Pe'er, D. (2005). Bayesian network analysis of signaling networks: A primer. *Science's STKE, 281*, 14.

Pe'er, D., & Hartemink, A. (2004). Single-cell gene expression analysis: Implications for neurodegenerative and neuropsychiatric disorders. *Neurochemical Research, 29*, 1053–1064. doi:10.1023/B:NERE.0000023593.77052.f7

Pe'er, D., Regev, A., & Tanay, A. (2002). Minreg: Inferring an active regulator set. *Bioinformatics (Oxford, England), 18*(Suppl. 1), 258–267.

Pe'er, D., Regev, A., Elidan, G., & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics (Oxford, England), 17*(Suppl 1), S215–S224.

Pe'er, D., Tanay, A., & Regev, A. (2006). MinReg: A scalable algorithm for learning parsimonious regulatory networks in yeast and mammals. *Journal of Machine Learning Research, 7*, 167–189.

Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving.* Reading, MA: Addison-Wesley.

Pearl, J. (1987). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence, 32*, 245–257. doi:10.1016/0004-3702(87)90012-9

Pearl, J. (1998). *Probabilistic reasoning in intelligent systems.* San Francisco: Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, reasoning, and inference.* Cambridge University Press.

Pearl, J., & Verma, T. S. (1990). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence* (pp. 220-227).

Peck, S. C. (2005). Update on proteomics in Arabidopsis. Where do we go from here? *Plant Physiology, 138*(2), 591–599. doi:10.1104/pp.105.060285

Pedraza, J. M., & van Oudenaarden, A. (2005). Noise propagation in gene networks. *Science, 307*(5717), 1965–1969. doi:10.1126/science.1109090

Peifer, M., & Timmer, J. (2007). Parameter estimation in ordinary differential equations for biochemical processes using the method of multiple shooting. *IET Systems Biology, 1*(2), 78–88. doi:10.1049/iet-syb:20060067

Peleg, M., Rubin, D., & Altman, R. B. (2005). Using Petri net tools to study properties and dynamics of biological systems. *Journal of the American Medical Informatics Association, 12*(2), 181–199. doi:10.1197/jamia.M1637

Peleš, S., Munsky, B., & Khammash, M. (2006). Reduction and solution of the chemical master equation using time scale separation and finite state projection. *The Journal of Chemical Physics*, 125.

Pellegrino, M., Provero, P., Silengo, L., & Di Cunto, F. (2004). CLOE: Identification of putative functional relationships among genes by comparison of expression profiles between two species. *BMC Bioinformatics, 5*, 179. doi:10.1186/1471-2105-5-179

Peng, H., Long, F., Zhou, J., Leung, G., Eisen, M., & Myers, E. (2007). Automatic image analysis for gene expression patterns of fly embryos. *BMC Cell Biology, 8*(Suppl 1), S7. doi:10.1186/1471-2121-8-S1-S7

Peng, X., Zhou, W., & Wang, Y. (2007). Efficient binomial leap method for simulating chemical kinetics. *The Journal of Chemical Physics, 126*, 224109. doi:10.1063/1.2741252

Pérez-Enciso, M., Quevedo, J. R., & Bahamonde, A. (2007). Genetical genomics: Use all data. *BMC Genomics, 8*(69).

Perrin, B. E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., & d'Alché-Buc, F. (2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics (Oxford, England), 19*(Suppl. 2), 38–48. doi:10.1093/bioinformatics/btg1071

Peterson, J. L. (1981). *Petri net theory and the modeling of systems.* Englewood Cliffs, NJ: Prentice Hall.

Petri, C. A. (1962). *Communication with automata (in German).* TU Darmstadt, Darmstadt.

Pieroni, E., de la Fuente van Bentem, S., Mancosu, G., Capobianco, E., Hirt, H., & de la Fuente, A. (2008). Protein networking: Insights into global functional organization of proteomes. *Proteomics, 8*(4), 799–816. doi:10.1002/pmic.200700767

Pilpel, Y., Sudarsanam, P., & Church, G. M. (2001). Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics, 29*(2), 153–159. doi:10.1038/ng724

Plummer, M., Best, N., Cowles, K., & Vines, K. (2006). Coda: Convergence diagnosis and output analysis for MCMC. *R News, 6*(1), 7-11. Retrieved from http://CRAN.R-project.org/doc/Rnews/

Poland, F. (2008). Cell illustrator. Retrieved from http://www.fqs.pl/life_science/cell_illustrator

Popova-Zeugmann, L., Heiner, M., & Koch, I. (2005). Modelling and analysis of biochemical networks with time Petri nets. *Fundamenta Informaticae, 67,* 149–163.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2002). *Numerical recipes in C++*. Cambridge: Cambridge University Press.

Proth, J.-M., & Xie, X. (1997). *Petri nets: A tool for design and management of manufacturing systems.* John Wiley & Sons, Inc.

Ptashne, M. (1986). *A genetic switch, gene control, and phage lambda*. Cambridge, MA: Cell Press.

Ptashne, M. (1992). *A genetic switch: Phage λ and higher organisms.* Cambridge, MA: Blackwell Scientific Publications and Cell Press.

Ptashne, M. (2004). *A genetic switch* (3rd ed.). Cold Spring Harbor Laboratory Press.

Ptashne, M., & Gann, A. (2002). *Genes and signals.* Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press.

Purdom, E., & Holmes, S. P. (2005). Error distribution for gene expression data. *Statistical Applications in Genetics and Molecular Biology, 4*(1), 16. doi:10.2202/1544-6115.1070

R Development Core Team. (2006). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing.

Radde, N., & Kaderali, L. (2007). Bayesian inference of gene regulatory networks using gene expression data. (LNBI 4414), Bird 07, Springer series, 1-15.

Raftery, A. E. (1993). Bayesian model selection in structural equation models. In K. A. Bollen & J. S. Long (Eds.), *Testing structural equation models* (pp. 163-180). Beverly Hills: Sage.

Rahmel, J. (1996). SplitNet: A dynamic hierarchical network model. *AAAI/IAAI,* 2, 1404.

Ram, R., & Chetty, M. (2007). A guided genetic algorithm for gene regulatory network. *Proc IEEE Congress on Evolutionary Computation* (pp. 3862-3869).

Ram, R., & Chetty, M. (2007). Framework for path analysis for learning gene regulatory network. *Pattern Recognition in Bioinformatics, Springer* (pp. 264-273).

Ram, R., & Chetty, M. (2007). Learning structure of gene regulatory networks. *6th IEEE International Conference on Computer and Information Science* (pp. 525-531).

Ram, R., & Chetty, M. (2008). Generating synthetic gene regulatory networks. *Pattern Recognition in Bioinformatics, Springer* (pp. 237-249).

Ram, R., Chetty, M., & Dix, T. I. (2006). Causal modelling of gene regulatory network. *Proc. IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, (CIBCB)* (pp. 1-8).

Ram, R., Chetty, M., & Dix, T. I. (2006). Fuzzy model for gene regulatory networks. *Proc. IEEE Congress on Evolutionary Computation* (pp. 1450-1455).

Rao, C. V., & Arkin, A. P. (2003). Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. *Journal of Physical Chemistry, 118*(11).

Rao, C. V., Wolf, D. M., & Arkin, A. P. (2002). Control, exploitation, and tolerance of intracellular noise. *Nature, 420,* 231–238. doi:10.1038/nature01258

Raser, J. M., & O'Shea, E. K. (2004). Control of stochasticity in eukaryotic gene expression. *Science, 304,* 1811–1814. doi:10.1126/science.1098641

Raser, J. M., & O'Shea, E. K. (2005). Noise in gene expression: Origins, consequences, and control. *Science, 309*(5743), 2010–2013. doi:10.1126/science.1105891

Rashkovsky, I., & Margaliot, M. (2007). Nicholson's blowflies revisited: A fuzzy modeling approach. *Fuzzy Sets and Systems, 158*, 1083–1096. doi:10.1016/j.fss.2006.11.001

Rastegar, M., Lemaigre, F. P., & Rousseau, G. G. (2000). Control of gene expression by growth hormone in liver: Key role of a network of transcription factors. *Molecular and Cellular Endocrinology, 164*(1-2), 1–4. doi:10.1016/S0303-7207(00)00263-X

Rathinam, M., Cao, Y., Petzold, L., & Gillespie, D. (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics*, *119*, 12784–12794. doi:10.1063/1.1627296

Rathinam, M., Petzold, L. R., & Cao, Y. (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *The Journal of Chemical Physics, 119*, 12784. doi:10.1063/1.1627296

Ravasz, E., & Barabási, A. L. (2003). Hierarchical organization in complex networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *67*(2). doi:10.1103/PhysRevE.67.026112

Rechenberg, I. (1973). *Evolutionsstrategie-optimierung technischer systeme nach prinzipien der biologischen evolution*. Unpublished doctoral dissertation. Reprinted by Frommann-Holzboog Verlag, Stuttgart-Bad Cannstatt.

Reddy, V. N. (1994). *Modeling biological pathways: A discrete event systems approach*. University of Maryland.

Reddy, V. N., Mavrovouniotis, M. L., & Liebman, M. N. (1993). *Petri net representations in metabolic pathways.* Paper presented at the ISMB International Conference on Intelligent Systems in Molecular Biology Bethesda, MD.

Reddy, V. N., Mavrovouniotis, M. L., & Liebman, M. N. (1996). Qualitative analysis of biochemical reaction systems. *Computers in Biology and Medicine*, *26*(1), 9–24. doi:10.1016/0010-4825(95)00042-9

Reece-Hoyes, J. S., Deplancke, B., Shingles, J., Grove, C. A., Hope, I. A., & Walhout, A. J. (2005). A compendium of *Caenorhabditis elegans* regulatory transcription factors: A resource for mapping transcription regulatory networks. *Genome Biology, 6*(13), R110. doi:10.1186/gb-2005-6-13-r110

Refanidis, I., & Vlahavas, I. P. (2003). Multiobjective heuristic state-space planning. *Artificial Intelligence*, *145*(1-2), 1–32. doi:10.1016/S0004-3702(02)00371-5

Reil, T. (1999) Dynamics of gene expression in an artificial genome-implications for biological and artificial ontogeny. In D. Floreano, J.-D. Nicoud & F. Mondada (Eds.), *Advances in Artificial Life: 5th European Conference* (pp. 457-466). Lausanne, Switzerland: Springer Verlag.

Reinitz, J., & Vaisnys, J. R. (1990). Theoretical and experimental analysis of the phage lambda genetic switch implies missing levels of cooperativity. *Journal of Theoretical Biology, 145*, 295–318. doi:10.1016/S0022-5193(05)80111-0

Reisig, W. (1985). *Petri nets: An introduction* (2nd ed., Vol. 4). Berlin Springer-Verlag.

Ren, B., Robert, F., Wyrick, J. J., Aparicio, O., Jennings, E. G., & Simon, I. (2000). Genome-wide location and function of DNA binding proteins. *Science*, *290*(5500), 2306–2309. doi:10.1126/science.290.5500.2306

Resat, V. H., Wiley, H. S., & Dixon, D. A. (2001). Probability-weighted dynamic Monte Carlo method for reaction kinetics simulations. *The Journal of Physical Chemistry B, 105*, 11026–11034. doi:10.1021/jp011404w

Resh, M. D. (2006). Trafficking and signaling by fatty-acylated and prenylated proteins. *Nature Chemical Biology*, *2*(11), 584–590. doi:10.1038/nchembio834

Ribeiro, A. S. (2007). Effects of coupling strength and space on the dynamics of coupled toggle switches in stochastic gene networks with multiple-delayed reactions. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *75*(1).

Ribeiro, A. S., & Kauffman, S. A. (2007). Noisy attractors and ergodic sets in models of genetic regulatory networks. *Journal of Theoretical Biology*, *247*(4), 743–755. doi:10.1016/j.jtbi.2007.04.020

Ribeiro, A. S., & Lloyd-Price, J. (2007). SGNSim, a stochastic genetic networks simulator. *Bioinformatics (Oxford, England), 23*(6), 777–779. doi:10.1093/bioinformatics/btm004

Ribeiro, A. S., Charlebois, D., Lloyd-Price, J., & Kauffman, S.A. (2006, May 10-12). IADGRN: Inferring gene regulatory networks from time series of genes activity. Increasing the scope of efficiency to more general interaction functions between genes and more complex time

series. *6th Int. Conf. Canadian Proteomics Initiative, CPI*, Edmonton, Alberta, Canada.

Ribeiro, A. S., Zhu, R., & Kauffman, S. A. (2006). A general modeling strategy for gene regulatory networks with stochastic dynamics. *Journal of Computational Biology, 13*(9), 1630–1639. doi:10.1089/cmb.2006.13.1630

Richardson, T. (1996). A discovery algorithm for directed cyclic graphs. In *Proceedings of the 12th international conference on uncertainty in artificial intelligence* (pp. 454-461).

Richardson, T. (1996). *A polynomial-time algorithm for deciding Markov equivalence of directed cyclic graphical models.* Paper presented at the Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence, Portland, Oregon.

Richardson, T., & Spirtes, P. (1999). Automated discovery of linear feedback models. In C. Glymour & G. F. Cooper (Eds.), *Computation, causation, and discovery* (pp. 253-304). Cambridge, MA: MIT Press.

Riechmann, J. L., Heard, J., Martin, G., Reuber, L., Jiang, C., & Keddie, J. (2000). *Arabidopsis* transcription factors: Genomewide comparative analysis among eukaryotes. *Science, 290*(5499), 2105–2110. doi:10.1126/science.290.5499.2105

Riggelsen, C. (2006). *Approximation methods for efficient learning of Bayesian networks.* Unpublished doctoral dissertation, Utrecht University.

Rissanen, J. (2007). *Information and complexity in statistical modeling.* New York: Springer.

Robertson, K. D. (2002). DNA methylation and chromatin–unravelling the tangled web. *Oncogene, 21*(35), 5361–5379. doi:10.1038/sj.onc.1205609

Robinson, R. W. (1976). Counting unlabeled acyclic digraphs. In *Proc. Fifth Australian Conf. Combinatorial Math.* (pp. 28-43).

Rockman, M. V. (2008). Reverse engineering the genotype-phenotype map with natural genetic variation. *Nature, 456*(7223), 738–744. doi:10.1038/nature07633

Rockman, M. V., & Kruglyak, L. (2006). Genetics of global gene expression. *Nature Reviews. Genetics, 7*(11), 862–872. doi:10.1038/nrg1964

Rodrigo, G., Carrera, J., & Jaramillo, A. (2007). Genetdes: Automatic design of transcriptional networks. *Bioinformatics (Oxford, England), 23*(14), 1857–1858. doi:10.1093/bioinformatics/btm237

Rogers, S., & Girolami, M. (2005). A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics (Oxford, England), 21*(14), 3131–3137. doi:10.1093/bioinformatics/bti487

Ronald, J., Brem, R. B., Whittle, J., & Kruglyak, L. (2005). Local regulatory variation in *Saccharomyces cerevisiae*. *PLOS Genetics, 1*(2), e25. doi:10.1371/journal.pgen.0010025

Rosa, G. J., de Leon, N., & Rosa, A. J. (2006). Review of microarray experimental design strategies for genetical genomics studies. *Physiological Genomics, 28*(1), 15–23. doi:10.1152/physiolgenomics.00106.2006

Rosenfeld, N., Young, J. W., Alon, U., Swain, P. S., & Elowitz, M. B. (2005). Gene regulation at the single-cell level. *Science, 307*, 1962–1965. doi:10.1126/science.1106914

Rostoks, N., Borevitz, J. O., Hedley, P. E., Russell, J., Mudie, S., & Morris, J. (2005a). Single-feature polymorphism discovery in the barley transcriptome. *Genome Biology, 6*(6), R54. doi:10.1186/gb-2005-6-6-r54

Rostoks, N., Mudie, S., Cardle, L., Russell, J., Ramsay, L., & Booth, A. (2005b). Genome-wide SNP discovery and linkage analysis in barley based on genes responsive to abiotic stress. *Molecular Genetics and Genomics, 274*(5), 515–527. doi:10.1007/s00438-005-0046-z

Roth, R. B., Hevezi, P., Lee, J., Willhite, D., Lechner, S. M., Foster, A. C., & Zlotnik, A. (2006). Gene expression analyses reveal molecular relationships among 20 regions of the human CNS. *Neurogenetics, 7*, 67–80. doi:10.1007/s10048-006-0032-6

Roth, S. Y., Denu, J. M., & Allis, C. D. (2001). Histone acetyltransferases. *Annual Review of Biochemistry, 70*, 81–120. doi:10.1146/annurev.biochem.70.1.81

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics, 20*, 53–65. doi:10.1016/0377-0427(87)90125-7

Roussel, M. R. (1996). The use of delay differential equations in chemical kinetics. *Journal of Physical Chemistry, 100*, 8323–8330. doi:10.1021/jp9600672

Roussel, M., & Zhu, R. (2006). Validation of an algorithm for delay stochastic simulation of transcription and translation in prokaryotic gene expression. *Physical Biology, 3*, 274–284. doi:10.1088/1478-3975/3/4/005

Roxin, A., Riecke, H., & Solla, S. A. (2004). Self-sustained activity in a small-world network of excitable neurons. *Physical Review Letters, 92*, 198101. doi:10.1103/PhysRevLett.92.198101

Roy, S. N. (1957). *Some aspects of multivariate analysis.* New York: Wiley.

Rozanov, D. V., D'Ari, R., & Sineoky, S. P. (1998). RecA-independent pathways of lambdoid prophage induction in Escherichia coli. *Journal of Bacteriology, 180*, 6306–6315.

Rozin, V., & Margaliot, M. (2007). The fuzzy ant. *IEEE Computational Intelligence Magazine, 2*, 18–28. doi:10.1109/MCI.2007.906684

Rubinstein, R. Y., & Kroese, D. P. (2004). *The cross entropy method.* Springer Verlag.

Rudge, T., & Geard, N. (2005). Evolving gene regulatory networks for cellular morphogenesis. In *Recent Advances in Artificial Life, Advances in Natural Computation,* 3, 231-252. World Scientific Publishers.

Rzhetsky, A., & Gomez, S. M. (2001). Birth of scale-free molecular networks and the number of distinct DNA and protein domains per genome. *Bioinformatics (Oxford, England), 17*(10), 988–996. doi:10.1093/bioinformatics/17.10.988

Sabatti, C., & James, G. M. (2006). Bayesian sparse hidden components analysis for transcription regulation networks. *Bioinformatics (Oxford, England), 22*(6), 739–746. doi:10.1093/bioinformatics/btk017

Sackmann, A. (2005). *Modeling and simulation of signal transduktion pathways in Saccharomyces cerevisiae based on Petri net theory (in German).* Technical University of Applied Sciencesm, Ernst-Moritz-Arndt-University Greifswald, Berlin, Greifswald.

Sackmann, A., Formanowicz, D., Formanowicz, P., Koch, I., & Blazewicz, J. (2007). An analysis of the Petri net based model of the human body iron homeostasis process. *Computational Biology and Chemistry, 31*, 1–10. doi:10.1016/j.compbiolchem.2006.09.005

Sackmann, A., Heiner, M., & Koch, I. (2006). Application of Petri net based analysis techniques to signal transduction pathways. *Journal, 7*, 482. doi:.doi:10.1186/1471-2105-7-482

Saito, A., Nagasaki, M., Doi, A., Ueno, K., & Miyano, M. (2006). Cell fate simulation model of gustatory neurons with microRNAs double-negative feedback loop by hybrid functional Petri net with extension. *Genome Informatics, 17*(1), 100–111.

Saitou, N., & Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, *4*, 406–425.

Sakamoto, E., & Iba, H. (2001). Inferring a system of differential equations for a gene regulatory network by using genetic programming. In *Proceedings of the 2001 Congress on Evolutionary Computation*, *1*, 720-726. IEEE-Press.

Salgado, H., Gama-Castro, S., Martinez-Antonio, A., Diaz-Peredo, E., Sanchez-Solano, F., & Peralta-Gil, M. (2004). RegulonDB (version 4.0): Transcriptional regulation, operon organization and growth conditions in Escherichia coli K-12. *Nucleic Acids Research, 32*(Database issue), D303–D306. doi:10.1093/nar/gkh140

Salgado, H., Gama-Castro, S., Peralta-Gil, M., Diaz-Peredo, E., Sanchez-Solano, F., & Santos-Zavaleta, A. (2006). RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research, 34*, D394–D397. doi:10.1093/nar/gkj156

Salgado, H., Santos, A., Garza-Ramos, U., van Helden, J., Diaz, E., & Collado-Vides, J. (1999). RegulonDB (version 2.0): A database on transcriptional regulation in Escherichia coli. *Nucleic Acids Research, 27*(1), 59–60. doi:10.1093/nar/27.1.59

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millan-Zarate, D., Blattner, F. R., & Collado-Vides, J. (2000). RegulonDB (version 3.0): Transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research, 28*(1), 65–67. doi:10.1093/nar/28.1.65

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millan-Zarate, D., Diaz-Peredo, E., & Sanchez-Solano, F. (2001). RegulonDB (version 3.2): Transcriptional regulation and operon organization in Escherichia coli K-12. *Nucleic Acids Research, 29*(1), 72–74. doi:10.1093/nar/29.1.72

Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Peralta-Gil, M., Penaloza-Spinola, M. I., & Martinez-Antonio, A. (2006). The comprehensive updated regulatory network of Escherichia coli K-12. *BMC Bioinformatics, 7*, 5. doi:10.1186/1471-2105-7-5

Samoilov, M. (2003). Stochastic effects in enzymatic biomolecular systems: Framework, fast and slow species, and quasi-steady state approximations. In *Workshop on Dynamical Stochastic Modeling in Biology.*

Samoilov, M. S., & Arkin, A. P. (2006). Deviant effects in molecular reaction pathways. *Nature Biotechnology, 24*, 1235–1240. doi:10.1038/nbt1253

Samoilov, M., Plyasunov, S., & Arkin, A. P. (2005). Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. *Proceedings of the National Academy of Sciences US, 102*(7), 2310–2315. doi:10.1073/pnas.0406841102

Sanguinetti, G., Lawrence, N. D., & Rattray, M. (2006). Probabilistic inference of transcription factor concentrations and gene-specific regulatory activities. *Bioinformatics (Oxford, England), 22*(22), 2775–2781. doi:10.1093/bioinformatics/btl473

Santillan, M., & Mackey, M. C. (2001). Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data. *Proceedings of the National Academy of Sciences of the United States of America, 98*(4), 1364–1369. doi:10.1073/pnas.98.4.1364

Santillan, M., & Mackey, M. C. (2004). Why the lysogenic state of phage λ is so stable: A mathematical modeling approach. *Biophysical Journal, 86*, 75–84. doi:10.1016/S0006-3495(04)74085-0

Sasaki, M., Takeda, E., Takano, K., Yomogida, K., Katahira, J., & Yoneda, Y. (2005). *Genomics, 85*(5), 641–653. doi:10.1016/j.ygeno.2005.01.003

Savageau, M. (1976). *Biochemical systems analysis: A study of function and design in molecular biology.* Reading: Addison-Wesley.

Savageau, M. A. (1998). Rules for the evolution of gene circuitry. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing, 3*, 55–65.

Savageau, M., & Alves, R. (2006, July 31- August 4). Mathematical representation and controlled comparison of biochemical systems. Tutorial at the *International Conference on Molecular Systems Biology (ICMSB),* Munich, Germany.

Sayyed-Ahmad, A., Tuncay, K., & Peter, J. (2007). Transcriptional regulatory network refinement and quantification through kinetic modeling, gene expression microarray data, and information theory. *BMC Bioinformatics, 8*, 20. doi:10.1186/1471-2105-8-20

Scannell, D. R., & Wolfe, K. (2004). Rewiring the transcriptional regulatory circuits of cells. *Genome Biology, 5*(2), 206. doi:10.1186/gb-2004-5-2-206

Schadt, E. E., Monks, S. A., Drake, T. A., Lusis, A. J., Che, N., & Colinayo, V. (2003). Genetics of gene expression surveyed in maize, mouse and man. *Nature, 422*(6929), 297–302. doi:10.1038/nature01434

Schadt, E., Lamb, J., Yang, X., Zhu, J., Edwards, S., & Guhathakurta, D. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics, 37*(7), 710. doi:10.1038/ng1589

Schäfer, J., & Strimmer, K. (2005). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics (Oxford, England), 21*, 754–764. doi:10.1093/bioinformatics/bti062

Schäfer, J., & Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology, 4*, 32. doi:10.2202/1544-6115.1175

Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science, 270*(5235), 467–470. doi:10.1126/science.270.5235.467

Schilling, C. H., C. M. W., Famili, I., Church, G. M., Edwards, J. S., & Palsson, B.O. (2002). Genome-scale metabolic model of Helicobacter pylori 26695. *Journal of Bacteriology, 184*, 4582–4593. doi:10.1128/JB.184.16.4582-4593.2002

Schilling, C. H., Letscher, D., & Palsson, B. O. (2000). Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway oriented perspective. *Journal of Theoretical Biology, 203*, 229–248. doi:10.1006/jtbi.2000.1073

Schilstra, M. (2002). NetBuilder software. Retrieved from http://strc.herts.ac.uk/bio/maria/

Schroeder, M. D., Pearce, M., Fak, J., Fan, H., Unnerstall, U., & Emberly, E. (2004). Transcriptional control in the segmentation gene network of *Drosophila*. *PLoS Biology*, *2*(9), E271. doi:10.1371/journal.pbio.0020271

Schuster, S., Hilgetag, C., & Schuster, R. (1993). *Determining elementary modes of functioning in biochemical reaction networks at steady-state*. Paper presented at the Second Gauss Symposium.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*(2), 461–464. doi:10.1214/aos/1176344136

Schwefel, H. P. (1995). *Evolution and optimum seeking.* Wiley & Sons, Inc.

Schwikowski, B., Uetz, P., & Fields, S. (2000). A network of protein-protein interactions in yeast. *Nature Biotechnology*, *18*(12), 1257–1261. doi:10.1038/82360

Scott, M. P., & Carroll, S. B. (1987). The segmentation and homeotic gene network in early Drosophila development. *Cell*, *51*(5), 689–698. doi:10.1016/0092-8674(87)90092-4

Segal, E., Barash, Y., Simon, I., Friedman, N., & Koller, D. (2002). From promoter sequence to expression: A probabilistic framework. In *Proc. Sixth Annual Inter. Conf. on Computational Molecular Biology (RECOMB).*

Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U., & Gaul, U. (2008). Predicting expression patterns from regulatory sequence in Drosophila segmentation. *Nature*, *451*(7178), 535–540. doi:10.1038/nature06496

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., & Koller, D. (2003). Module networks: Identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, *34*(2), 166–176. doi:10.1038/ng1165

Segal, E., Wang, H., & Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics (Oxford, England)*, *19*(Suppl. 1), i264–i272. doi:10.1093/bioinformatics/btg1037

Segal, E., Yelensky, R., & Koller, D. (2003b). Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics (Oxford, England)*, *19*(Suppl. 1), 273–282. doi:10.1093/bioinformatics/btg1038

Segal, M. R. (2003). Regression approaches for microarray data analysis. *Journal of Computational Biology*, *10*(6), 961–980. doi:10.1089/106652703322756177

Segel, Lee A., & Slemrod, M. (1989). The quasi-steady-state assumption: A case study in perturbation. *SIAM Review*, *31*(3), 446–477. doi:10.1137/1031091

Serra, R., Villani, M., & Semeria, A. (2004). Genetic network models and statistical properties of gene expression data in knock-out experiments. *Journal of Theoretical Biology*, *227*, 149–157. doi:10.1016/j.jtbi.2003.10.018

Serra, R., Villani, M., Graudenzi, A., & Kauffman, S. A. (2007). Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. *Journal of Theoretical Biology*, *246*, 449–460. doi:10.1016/j.jtbi.2007.01.012

Sha, N., & Vanucci, M. (2004). Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics*, *60*, 812–819. doi:10.1111/j.0006-341X.2004.00233.x

Shea, M. A., & Ackers, G. K. (1985). The OR control system of bacteriophage lambda. A physical-chemical model for gene regulation. *Journal of Molecular Biology*, *181*, 211–230. doi:10.1016/0022-2836(85)90086-5

Shehadeh, L., Liebovitch, L. S., & Jirsa, V. K. (2006). The structure of genetic networks determined from mRNA levels measured by cDNA microarrays . *Physica A*, *364*, 297–314. doi:10.1016/j.physa.2005.08.069

Shen-Orr, S. S., Milo, R., Mangan, S., & Alon, U. (2002). Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, *31*(1), 64–68. doi:10.1038/ng881

Sherlock, G., Hernandez-Boussard, T., Kasarskis, A., Binkley, G., Matese, J. C., & Dwight, S. S. (2001). The Stanford microarray database. *Nucleic Acids Research*, *29*(1), 152–155. doi:10.1093/nar/29.1.152

Shimoni, Y., Friedlander, G., Hetzroni, G., Niv, G., Altuvia, S., & Biham, O. (2007). Regulation of gene expression by small non-coding RNAs: A quantitative view. *Molecular Systems Biology*, *3*, 138. doi:10.1038/msb4100181

Shipley, B. (2002). *Cause and correlation in biology: A user's guide to path analysis, structural equations, and causal inference.* Cambridge University Press.

Shmulevich, I., & Dougherty, E. R. (2003). Mappings between probabilistic Boolean networks. *Signal Processing*, *83*(4), 799–809. doi:10.1016/S0165-1684(02)00480-2

Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002). From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, *90*(11), 1778–1792. doi:10.1109/JPROC.2002.804686

Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics (Oxford, England)*, *18*(2), 261–274. doi:10.1093/bioinformatics/18.2.261

Shmulevich, I., Lähdesmäki, H., Dougherty, E. R., Astola, J., & Zhang, W. (2003b). The role of certain post classes in Boolean network models of genetic networks. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(19), 10734–10739. doi:10.1073/pnas.1534782100

Shoemaker, B. A., & Panchenko, A. R. (2007). Deciphering protein-protein interactions. Part I. Experimental techniques and databases. *PLoS Computational Biology*, *3*(3), e42. doi:10.1371/journal.pcbi.0030042

Shrager, J., Langley, P., & Pohorille, A. (2002). Guiding revision of regulatory models with expression data. *Pacific Symposium on Biocomputing*, *7*, 486–497.

Shults, B., & Kuipers, B. (1997). Proving properties of continuous systems: Qualitative simulation and temporal logic. *Artificial Intelligence*, *92*(1-2), 91–129. doi:10.1016/S0004-3702(96)00050-1

Shultzaberger, R. K., Chiang, D. Y., Moses, A. M., & Eisen, M. B. (2007). Determining physical constraints in transcriptional initiation complexes using DNA sequence analysis. *PLoS ONE*, *2*(11). doi:10.1371/journal.pone.0001199

Siler, W., & Buckley, J. J. (2004). *Fuzzy expert systems and fuzzy reasoning.* Wiley-Interscience.

Simao, E., Remy, E., Thieffry, D., & Chaouiya, C. (2005). Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in E.Coli. *Bioinformatics (Oxford, England)*, *21*(Suppl. 2), ii190–ii196. doi:10.1093/bioinformatics/bti1130

Singh, M., & Valtorta, M. (1993). An algorithm for the construction of (Bayesian) network structures from data. *The Ninth Conference on Uncertainty in Artificial Intelligence* (pp. 259-265). Washington, D.C.: Morgan Kaufmann.

Širava, M., Schäfer, T., Eiglsperger, M., Kaufmann, M., Kohlbacher, O., Bornberg-Bauer, E., & Lenhof, H. P. (2002). BioMiner-modeling, analyzing, and visualizing biochemical pathways and networks. *Bioinformatics (Oxford, England)*, *18*(Suppl. 2), S219–S230.

Smith, B. J. (2007). Boa: An r package for MCMC output convergence assessment and posterior inference. *Journal of Statistical Software*, *21*(11).

Smith, M., & Kohn, R. (1997). A Bayesian approach to nonparametric bivariate regression. *Journal of the American Statistical Association*, *92*, 1522–1535. doi:10.2307/2965423

Somogyi, R., & Sniegoski, C. (1996). Modeling the complexity of gene networks: Understanding multigenic and pleiotropic regulation. *Complexity*, *1*, 45–63.

Soranzo, N., Bianconi, G., & Altafini, C. (2007). Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: Synthetic vs. real data. *Bioinformatics (Oxford, England)*, *23*(13), 1640–1647. doi:10.1093/bioinformatics/btm163

Sosman, J. A., Weeraratna, A. T., & Sondak, V. K. (2004). When will melanoma vaccines be proven effective? *Journal of Clinical Oncology*, *22*(3), 387–389. doi:10.1200/JCO.2004.11.950

Sousa, J. M. C., & Kaymak, U. (2002). *Fuzzy decision making in modeling and control*. World Scientific.

Spellman, P. T., Sherlock, G., Zhang, M. Q., & Iyer, V. R. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, *9*(12), 3273–3297.

Spiegelhalter, D. J., Thomas, A., & Best, N. G. (1996). Computation on Bayesian graphical models. In J. M. Bernardo, J. O. Berger, A. P. Dawid & A. F. M. Smith (Eds.), *Bayesian statistics*, 5, 407-425.

Spirtes, P., & Meek, C. (1995). Learning Bayesian networks with discrete variables from data. In U. M. Fayyad & R. Uthurusamy (Eds.), *Proceedings of the first international conference on knowledge discovery and data mining* (pp. 294-299). San Jose, CA: AAAI Press.

Spirtes, P., Glymour, C., & Scheines, R. (2001). *Causation, Prediction and Search*. 2nd edition, Cambridge, MA: The MIT Press.

Spirtes, P., Glymour, C., Scheines, R., Kauffman, S., Aimale, V., & Wimberly, F. (2000). *Constructing Bayes-*

ian network models of gene expression networks from microarray data. Paper presented at the Proc. Atlantic Symp. Comp. Biol., Genome Inf. Syst., and Technol.

Sporns, O., & Tononi, G. (2007). Structural determinants of functional brain connectivity. In V. K. Jirsa & A. R. M. McIntosh (Eds.), *Handbook of brain connectivity*. Springer.

Srinivas, S. (1993). A generalization of the noisy-or model. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence*.

Srivastava, R., You, L., Summer, J., & Yin, J. (2002). Stochastic versus deterministic modeling of intracellular viral kinetics. *Journal of Theoretical Biology, 218*(3), 309–321. doi:10.1006/jtbi.2002.3078

Starke, P. H. (1990). *Analysis of Petri net models (in German).* Stuttgart: B. G. Teubner.

Steck, H., & Jaakkola, T. (2007). Predictive discretization during model selection. In *Proc. 11th International Conference on Artificial Intelligence and Statistics*.

Steggles, L. J., Banks, R., & Wipat, A. (2006). *Modelling and analysing genetic networks: From Boolean networks to Petri nets.* Paper presented at the Computational Methods in Systems Biology.

Steggles, L. J., Banks, R., & Wipat, A. (2007). Qualitatively modelling and analysing genetic regulatory networks: A Petri net approach. *Bioinformatics (Oxford, England), 23*(3), 336–343. doi:10.1093/bioinformatics/btl596

Stein, C. M., Song, Y., Elston, R. C., Jun, G., Tiwari, H. K., & Iyengar, S. K. (2003). Structural equation model-based genome scan for the metabolic syndrome. *BMC Genetics, 4*(Suppl 1), S99. doi:10.1186/1471-2156-4-S1-S99

Steiner, T., Olhofer, M., & Sendhoff, B. (2006). Towards shape and structure optimization with evolutionary development. In *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems, ALife X* (pp. 70-76).

Steiner, T., Schramm, L., Jin, Y., & Sendhoff, B. (2007). Emergence of feedback in artificial gene regulatory networks. In . *Proceedings of the Congress on Evolutionary Computation, 2007*, 867–874. doi:10.1109/CEC.2007.4424561

Steinhausen, D., & Langer, K. (1977). *Cluster analysis: An introduction to methods for automatic classification (in German).* Berlin: de Gruyter.

Steuer, R., Kurths, J., Daub, C. O., Weise, J., & Selbig, J. (2002). The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics (Oxford, England), 18*(Suppl 2), S231–S240.

Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains.* Princeton University Press.

Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences of the United States of America, 100*(16), 9440–9445. doi:10.1073/pnas.1530509100

Storey, J., Akey, J., & Kruglyak, L. (2005). Multiple locus linkage analysis of genomewide expression in yeast. *PLoS Biology, 3*(8), e267. doi:10.1371/journal.pbio.0030267

Stormo, G. D. (2000). DNA binding sites: Representation and discovery. *Bioinformatics (Oxford, England), 16*(1), 16–23. doi:10.1093/bioinformatics/16.1.16

Strogatz, S. H. (2001). Exploring complex networks. *Nature, 410*, 268–276. doi:10.1038/35065725

Strogatz, S. H. (Ed.). (2000). *Nonlinear dynamics and chaos. Studies in nonlinearity.* Westview Press.

Ström, A., Castella, P., & Rockwood, J. (1997). Mediation of NGF signaling by post-translational inhibition of HES-1, a basic helix–loop–helix repressor of neuronal differentiation. *Genes & Development, 11*, 3168–3181. doi:10.1101/gad.11.23.3168

Stuart, J. M., Segal, E., Koller, D., & Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic module. *Science, 302*(5643), 249–255. doi:10.1126/science.1087447

Studier, J. A., & Keppler, K. J. (1988). A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution, 5*, 729–731.

Stylianou, I. M., Korstanje, R., Li, R., Sheehan, S., Paigen, B., & Churchill, G. A. (2006). Quantitative trait locus analysis for obesity reveals multiple networks of interacting loci. *Mammalian Genome, 17*(1), 22–36. doi:10.1007/s00335-005-0091-2

Su, A. I., Wiltshire, T., Batalov, S., Lapp, H., Ching, K. A., & Block, D. (2004). A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America, 10*(16), 6062–6067. doi:10.1073/pnas.0400782101

Süel, G. M., Garcia-Ojalvo, J., Liberman, L. M., & Elowitz, M. B. (2006). An excitable gene regulatory circuit induces transient cellular differentiation. *Nature*, *440*, 545–550. doi:10.1038/nature04588

Süel, G. M., Kulkarni, R. P., Dworkin, J., Garcia-Ojalvo, J., & Elowitz, M. B. (2007). Tunability and noise dependence in differentiation dynamics. *Science*, *315*, 1717–1719. doi:10.1126/science.1137455

Sun, L., Blair, H. C., Peng, Y., Zaidi, N., Adebanjo, O. A., & Wu, X. B. (2005). Calcineurin regulates bone formation by the osteoblast. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 17130–17135. doi:10.1073/pnas.0508480102

Tabus, I., & Astola, J. (2001). On the use of the MDL principle in gene expression prediction. *Journal of Applied Signal Processing*, *4*, 297–303. doi:10.1155/S1110865701000270

Tabus, I., Rissanen, J., & Astola, J. (2002). Normalized maximum likelihood models for Boolean regression with application to prediction and classification in genomics. In W. Zhang & I. Shmulevich (Eds.), *Computational and statistical approaches to genomics*. Boston, MA: Kluver.

Tabus, I., Rissanen, J., & Astola, J. (2003). Classification and feature gene selection using the normalized maximum likelihood model for discrete regression. *Signal Processing*, *84*(4), 713–727. doi:10.1016/S0165-1684(02)00470-X

Tadesse, M. G., Vanucci, M., & Lio, P. (2004). Identification of DNA regulatory motifs using Bayesian variable selection. *Bioinformatics (Oxford, England)*, *2*(16), 2553–2561. doi:10.1093/bioinformatics/bth282

Takai-Igarashi, T. (2005). Ontology based standardization of Petri net modeling for signalling pathways. *Journal*, *5*, 0047

Tamada, Y., Bannai, H., Imoto, S., & Katayama, T. (2005). Utilizing evolutionary information and gene expression data for extimating gene networks with Bayesian network models. *Journal of Bioinformatics and Computational Biology*, *3*(6), 1295–1313. doi:10.1142/S0219720005001569

Tamada, Y., Kim, S., Bannai, H., & Imoto, S. (2003). Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics (Oxford, England)*, *19*(Suppl 2), II227–II236. doi:10.1093/bioinformatics/btg1082

Tan, W.-Y. (2002). *Stochastic models with applications to genetics, cancers, AIDS, and other biomedical systems*. World Scientific Publishing Co. Ltd.

Tavazoie, S., Hugues, J. D., Campell, M. J., Cho, R. J., & Church, G. M. (1999). Systematic determination of genetic network architecture. *Nature Genetics*, *22*, 281–285. doi:10.1038/10343

Tegner, J., Yeung, M. K., Hasty, J., & Collins, J. J. (2003). Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(10), 5944. doi:10.1073/pnas.0933416100

Teichmann, S. A., & Babu, M. M. (2004). Gene regulatory network growth by duplication. *Nature Genetics*, *36*(5), 492–496. doi:10.1038/ng1340

Teixeira, M. C., Monteiro, P., Jain, P., Tenreiro, S., Fernandes, A. R., & Mira, N. P. (2006). The YEASTRACT database: A tool for the analysis of transcription regulatory associations in Saccharomyces cerevisiae. *Nucleic Acids Research*, *34*(Suppl 1), D446–D451. doi:10.1093/nar/gkj013

Teixeira, M. C., Monteiro, P., Jain, P., Tenreiro, S., Fernandes, A. R., Mira, N. P., et al. (2006). The YEASTRACT database: A tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*, *Nucleic Acids Research*, 34, D446-D451. Oxford University Press. Retrieved in December 2007, from http://www.yeastract.com

TGI-group. (2008). *Petri Nets World*. Retrieved from http://www.informatik.uni-hamburg.de./TGI/PetriNets

Thaller, G., & Hoeschele, I. (2000). Fine-mapping of quantitative trait loci in half-sib families using current recombinations. *Genetical Research*, *76*, 87–104. doi:10.1017/S0016672300004638

Thanos, D., & Maniatis, T. (1995). Virus induction of human IFN beta gene expression requires the assembly of an enhanceosome. *Cell*, *83*, 1091–1100. doi:10.1016/0092-8674(95)90136-1

Thieffry, D. (2007). Dynamical roles of biological regulatory circuits. *Briefings in Bioinformatics*, *8*(4), 220–225. doi:10.1093/bib/bbm028

Thieffry, D., & Thomas, R. (1998). Qualitative analysis of gene networks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 77–88.

Thieffry, D., Huerta, A. M., Pèrez-Rueda, E., & Collado-Vides, J. (1998). From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in escherichia coli. *BioEssays*, *20*, 433–440. doi:10.1002/(SICI)1521-1878(199805)20:5<433::AID-BIES10>3.0.CO;2-2

Thomas, A., O'Hara, B., Ligges, U., & Sturtz, S. (2006). Making BUGS open. *R News*, *6*, 12–17.

Thomas, R. (1973). Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, *42*(3), 563–585. doi:10.1016/0022-5193(73)90247-6

Thomas, R. (1998). Laws for the dynamics of regulatory networks. *J. Dev. Biol.*, *42*, 479–485.

Thomas, R., & D'Ari, R. (1990). *Biological feedback*. Boca Raton, FL: CRC Press.

Thomas, R., & Kaufman, M. (2001). Multistationarity, the basis of cell differentiation and memory, II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos (Woodbury, N.Y.)*, *11*(1), 180–195. doi:10.1063/1.1349893

Thomas, R., Thieffry, D., & Kaufman, M. (1995). Dynamical behaviour of biological regulatory networks - I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, *57*(2), 247–276.

Thorsson, V., Hörnquist, M., Siegel, A. F., & Hood, L. (2005). Reverse engineering galactose regulation in yeast through model selection. *Statistical Applications in Genomics*, 4(1).

Threadgill, D. W. (2006). Meeting report for the 4th Annual Complex Trait Consortium meeting: From QTLs to systems genetics. *Mammalian Genome*, *17*(1), 2–4. doi:10.1007/s00335-005-0153-5

Tian, T., & Burrage, K. (2004). Binomial leap methods for simulating stochastic chemical kinetics. *The Journal of Chemical Physics*, *121*(21), 10356–10364. doi:10.1063/1.1810475

Tian, T., Burrage, K., Burrage, P. M., & Carletti, M. (2007). Stochastic delay differential equations for genetic regulatory networks. *Journal of Computational and Applied Mathematics*, *205*(2), 696–707. doi:10.1016/j.cam.2006.02.063

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B. Methodological*, *58*(1), 267–288.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S. Q., & Lewis, S. E. (2002). Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biology*, *3*(12). doi:10.1186/gb-2002-3-12-research0088

Tomassini, M., Giacobini, M., & Darabos, C. (2007). Performance and robustness of cellular automata computation on irregular networks. *Advances in Complex Systems*, *10*, 85–110. doi:10.1142/S0219525907001124

Tong, A. H. Y., Lesage, G., Bader, G. D., Ding, H., Xu, H., & Xin, X. (2004). Global mapping of the yeast genetic interaction network. *Science*, *303*(5659), 808–813. doi:10.1126/science.1091317

Torres, A., & Nieto, J. J. (2006). Fuzzy logic in medicine and bioinformatics. *Journal of Biomedicine & Biotechnology*, 1–7. doi:10.1155/JBB/2006/91908

Toulouse, T., Ao, P., Shmulevich, I., & Kauffman, S. A. (2005). Noise in a small genetic circuit that undergoes bifurcation. *Complexity*, *11*(1), 45–51. doi:10.1002/cplx.20099

Toyn, J. H. (1997). The Swi5 transcription factor of Saccharomyces cerevisiae has a role in exit from mitosis through induction of the Cdk-inhibitor Sic1 in telophase. *Genetics*, *145*, 85–96.

Tran, L. S., Nakashima, K., Shinozaki, K., & Yamaguchi-Shinozaki, K. (2007). Plant gene networks in osmotic stress response: From genes to regulatory networks. *Methods in Enzymology*, *428*, 109–128. doi:10.1016/S0076-6879(07)28006-1

Tran, N., Baral, C., & Shankland, C. (2005). Issues in reasoning about interaction networks in cells: Necessity of event ordering knowledge. In *Proceedings of Twentieth National Conference on Artificial Intelligence*, Pittsburgh, PA (pp. 676-682).

Travers, A., & Muskhelishvili, G. (2005). DNA supercoiling-a global transcriptional regulator for enterobacterial growth? *Nature Reviews Microbiology*, *3*, 157–169. doi:10.1038/nrmicro1088

Tron, E., & Margaliot, M. (2004). Mathematical modeling of observed natural behavior: A fuzzy logic approach. *Fuzzy Sets and Systems*, *146*, 437–450. doi:10.1016/j.fss.2003.09.005

Tron, E., & Margaliot, M. (2005). How does the Dendrocoleum lacteum orient to light? A fuzzy modeling approach.

Fuzzy Sets and Systems, *155*, 236–251. doi:10.1016/j.fss.2005.03.008

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., & Tibshirani, R. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics (Oxford, England)*, *17*(6), 520–525. doi:10.1093/bioinformatics/17.6.520

Tsai, H. K., Lu, H. H., & Li, W. H. (2005). Statistical methods for identifying yeast cell cycle transcription factors. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 13532–13537. doi:10.1073/pnas.0505874102

Tsang, J., Zhu, J., & van Oudenaarden, A. (2007). MicroRNA-mediated feedback and feedforward loops are recurrent network motifs in mammals. *Molecular Cell*, *26*(5), 753–767. doi:10.1016/j.molcel.2007.05.018

Tsantoulis, P. K., & Gorgoulis, V. G. (2005). Involvement of E2F transcription factor family in cancer. *European Journal of Cancer*, *41*(16), 2403–2414. doi:10.1016/j.ejca.2005.08.005

Tu, Z., Wang, L., Arbeitman, M. N., Chen, T., & Sun, F. (2006). An integrative approach for causal gene identification and gene regulatory pathway inference. *Bioinformatics (Oxford, England)*, *22*(14), e489–e496. doi:10.1093/bioinformatics/btl234

Tuglus, C., & van der Laan, M. (2008). FDR controlling procedure for mult-stage analysis. *U. C. Berkeley Div. Biostsat*. Working paper series, paper 239.

Turner, T., Schnell, S., & Burrage, K. (2004). Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry*, *28*, 165. doi:10.1016/j.compbiolchem.2004.05.001

Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(9), 5116–5121. doi:10.1073/pnas.091062498

Tuttle, L. M., Salis, H., Tomshine, J., & Kaznessis, Y. N. (2005). Model-driven designs of an oscillating gene network. *Biophysical Journal*, *89*, 3873–3883. doi:10.1529/biophysj.105.064204

Tyson, J. J., & Othmer, H. G. (1978). The dynamics of feedback control circuits in biochemical pathways. R. Rosen (Ed.) New York: Academic Press.

Tyson, J. J., Chen, K. C., & Novak, B. (2003). Sniffers, buzzers, toggles, and blinkers: Dynamics of regulatory and signaling pathways in the cell. *Current Opinion in Cell Biology*, *15*(2), 221–231. doi:10.1016/S0955-0674(03)00017-6

Uetz, P., Giot, L., Cagney, G., & Mansfield, T. A. (2000). A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, *403*, 623–627. doi:10.1038/35001009

Ulitsky, I., Gat-Viks, I., & Shamir, R. (2008). MetaReg: A platform for modeling, analysis, and visualization of biological systems using large-scale experimental data. *Genome Biol*., 2; 9(1): R1.

Vahedi, G., Faryabi, B., Chamberland, J.-F., Data, A., & Dougherty, E. R. (in press). Intervention in gene regulatory networks via a stationary mean-first-passage time control policy.

Vahedi, G., Ivanov, I., & Dougherty, E. R. (in press). Inference of Boolean networks under constraint on bidirectional gene relationships.

Vallabhajosyula, R. R., Chickarmane, V., & Sauro, H. M. (2006). Conservation analysis of large biochemical networks. *Bioinformatics (Oxford, England)*, *22*(3), 346–353. doi:10.1093/bioinformatics/bti800

Valmari, A. (1998). *In lectures in Petri nets I: The state explosion problem* (Vol. 1491). Berlin: Springer.

Van den Bulcke, T., Lemmens, K., Van de Peer, Y., & Marchal, K. (2006). Inferring transcriptional networks by mining omics data. *Current Bioinformatics*, *1*, 301. doi:10.2174/157489306777827991

van der Aalst, W. M. P., Desel, J., & Oberweis, A. (1999). *Business process management: Models, techniques, and empirical studies* (Vol. 1806). Berlin: Springer-Verlag

van Dijk, A. D., ter Braak, C. J., Immink, R. G., Angenent, G. C., & van Ham, R. C. (2008). Predicting and understanding transcription factor interactions based on sequence level determinants of combinatorial control. *Bioinformatics (Oxford, England)*, *24*(1), 26–33. doi:10.1093/bioinformatics/btm539

van Driel, M. A., Bruggeman, J., Vriend, G., Brunner, H. G., & Leunissen, J. A. (2006). A text-mining analysis of the human phenome. *European Journal of Human Genetics*, *14*(5), 535–542. doi:10.1038/sj.ejhg.5201585

van Kampen, N. G. (1992). *Stochastic processes in physics and chemistry.* Elsevier.

van Someren, E. P., Vaes, B. L. T., Steegenga, W. T., Sijbers, A. M., Dechering, K. J., & Reinders, M. J. T. (2006). Least absolute regression network analysis of the murine osteoblast differentation network. *Bioinformatics (Oxford, England), 22*(4), 477–484. doi:10.1093/bioinformatics/bti816

van Someren, E. P., Wessels, E., Backer, L. F. A., & Reinders, M. J. T. (2003). Multicriterion optimization for genetic network modelling. *Signal Processing, 83*, 763–775. doi:10.1016/S0165-1684(02)00473-5

van Someren, E. P., Wessels, L. F. A., & Reinders, M. J. T. (2000). Linear modeling of genetic networks from experimental data. In *Proceedings of the eighth international conference on intelligent systems for molecular biology* (pp. 355-366).

van Someren, E. P., Wessels, L. F. A., Backer, E., & Reinders, M. J. T. (2002). Genetic network modelling. *Pharmacogenomics, 3*(4), 1–19.

Van Someren, E. P., Wessels, L. F., et al. (2001). Genetic network models: A comparative study. *Proc. of SPIE, Micro-arrays: Optical Technologies and Informatics.*

vanden Berghen, F. (2007). The new ultraFast LARS engine with n-fold-cross-validation and ridge regression. Retrieved in December 2007, from http://www.applied-mathematics.net/

Vázquez, A., Dobrin, R., Sergi, D., Eckmann, J.-P., Oltvai, Z. N., & Barabàsi, A.-L. (2004). The topological relationships between the large-scale attributes and local interactions patterns of complex networks. *Proceedings of the National Academy of Sciences of the United States of America, 101*(52), 17940–17945. doi:10.1073/pnas.0406024101

Veiga, D. F., Vicente, F. F., Grivet, M., de la Fuente, A., & Vasconcelos, A. T. (2007). Genome-wide partial correlation analysis of Escherichia coli microarray data. *Genetics and Molecular Research, 6*(4), 730–742.

Velculescu, V. E., Zhang, L., Vogelstein, B., & Kinzler, W. (1995). Serial analysis of gene expression. *Science, 270*, 484–487. doi:10.1126/science.270.5235.484

Vermeirssen, V., Barrasa, M. I., Hidalgo, C. A., Babon, J. A., Sequerra, R., & Doucette-Stamm, L. (2007). Transcription factor modularità in a gene-centered C. elegans core neuronal protein-DNA interaction network. *Genome Research, 17*(7), 1061–1071. doi:10.1101/gr.6148107

Vidal, M., Braun, P., Chen, E., Boeke, J. D., & Harlow, E. (1996). Genetic characterization of a mammalian protein-protein interaction domain by using a yeast reverse two-hybrid system. *Proceedings of the National Academy of Sciences of the United States of America, 93*(19), 10321–10326. doi:10.1073/pnas.93.19.10321

Vogelstein, B., Lane, D., & Levine, A. (2000). Surfing the p53 network. *Nature, 408*, 307–310. doi:10.1038/35042675

Vohradský, J. (2001). Neural network model of gene expression. *The FASEB Journal, 15*, 846–854. doi:10.1096/fj.00-0361com

Voit, E. O., & Ferreira, A. (2000). Computational analysis of biochemical systems: A practical guide for biochemists and molecular biologists. Cambridge University Press.

von Bertalanffy, L. (1968). General system theory. New York: Braziller.

von Neumann, J. (2001). In A. H. Taub (Ed.), *J. von Neumann, Collected Works*, 5, 288. New York: Macmillan.

Voss, K., Heiner, M., & Koch, I. (2003). Steady state analysis of metabolic pathways using Petri nets. *In Silico Biology, 3*(3), 367–387.

Vu, T., & Vohradsky, J. (2007). Nonlinear differential equation model for quantification of transcriptional regulation applied to microarray data of Saccharomyces cerevisiae. *Nucleic Acids Research, 35*(1), 279–287. doi:10.1093/nar/gkl1001

Vuylsteke, M., van Eeuwijk, F., Van Hummelen, P., Kuiper, M., & Zabeau, M. (2005). Genetic analysis of variation in gene expression in Arabidopsis thaliana. *Genetics, 171*(3), 1267–1275. doi:10.1534/genetics.105.041509

Waaijenborg, S., Verselewel de Witt Hamer, P. C., & Zwinderman, A. H. (2008). Quantifying the association between gene expressions and DNA-markers by penalized canonical correlation analysis. *Stat Appl Genet Mol Biol, 7*, Article3.

Wagner, A. (2001). How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy steps. *Bioinformatics (Oxford, England), 17*(12), 1183–1197. doi:10.1093/bioinformatics/17.12.1183

Wagner, A., & Fell, D. A. (2001). The small world inside large metabolic networks. *Proceedings. Biological Sciences*, *268*(1478), 1803–1810. doi:10.1098/rspb.2001.1711

Wang, C. (2007). Variational Bayesian approach to canonical correlation analysis. *IEEE Transactions on Neural Networks*, *18*, 905–910. doi:10.1109/TNN.2007.891186

Wang, D., & Nettleton, D. (2006). Identifying genes associated with a quantitative trait or quantitative trait locus via selective transcriptional profiling. *Biometrics*, *62*(2), 504–514. doi:10.1111/j.1541-0420.2005.00491.x

Wang, M., Chen, Z., & Cloutier, S. (2007). A hybrid Bayesian network learning method for constructing gene networks. *Computational Biology and Chemistry*, *31*(5-6), 361–372. doi:10.1016/j.compbiolchem.2007.08.005

Wang, R. S., Wang, Y., Zhang, X. S., & Chen, L. (2007). Inferring transcriptional regulatory networks from high-throughput data. *Bioinformatics (Oxford, England)*, *23*(22), 3056–3064. doi:10.1093/bioinformatics/btm465

Wang, X. F., & Chen, G. R. (2003). Complex networks: Small-world, scale-free, and beyond. *IEEE Circuits and Systems Magazine*, *3*(1), 6–20. doi:10.1109/MCAS.2003.1228503

Wang, Y., Joshi, T., Zhang, X. S., Xu, D., & Chen, L. (2006). Inferring gene regulatory networks from multiple microarray datasets. *Bioinformatics (Oxford, England)*, *22*(19), 2413. doi:10.1093/bioinformatics/btl396

Wasserman, W. W., & Sandelin, A. (2004). Applied bioinformatics for the identification of regulatory elements. *Nature Reviews. Genetics*, *5*(4), 276–287. doi:10.1038/nrg1315

Watson, J., Geard, N., & Wiles, J. (2004). Towards more biological mutation operators in gene regulation studies. [Elsevier.]. *Bio Systems*, *76*, 239–248. doi:10.1016/j.biosystems.2004.05.016

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. doi:10.1038/30918

Weaver, D. C., Workman, C. T., & Stormo, G. D. (1999). Modeling regulatory networks with weight matrices. *Pacific Symposium on Biocomputing, 4*, 112-123.

Weeraratna, A. T., Jiang, Y., Hostetter, G., Rosenblatt, K., Duray, P., Bittner, M., & Trent, J. M. (2002). Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma. *Cancer Cell*, *1*(3), 279–288. doi:10.1016/S1535-6108(02)00045-4

Wei, G. H., Liu, D. P., & Liang, C. C. (2004). Charting gene regulatory networks: Strategies, challenges, and perspectives. *The Biochemical Journal*, *381*(1), 1–12. doi:10.1042/BJ20040311

Weiner, N. (1948). Cybernetics or control and communication in the animal and the machine. Cambridge, MA: MIT Press.

Weisberg, S. (1985). *Applied linear regression*. New York: John Wiley.

Weiss, R., Basu, S., Hooshangi, S., Kalmbach, A., Karig, D., Mehreja, R., & Netravali, I. (2003). Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural Computing*, *2*(1), 47–84. doi:10.1023/A:1023307812034

Werhli, A. V., & Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, *6*(1). doi:10.2202/1544-6115.1282

Werhli, A. V., Grzegorczyk, M., & Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics (Oxford, England)*, *22*(20), 2523–2531. doi:10.1093/bioinformatics/btl391

West, M. (2003). Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics*, *7*, 723–732.

West, M. A., Kim, K., Kliebenstein, D. J., van Leeuwen, H., Michelmore, R. W., & Doerge, R. W. (2007). Global eQTL mapping reveals the complex genetic architecture of transcript-level variation in Arabidopsis. *Genetics*, *175*(3), 1441–1450. doi:10.1534/genetics.106.064972

Westerhoff, H. V., & van Dam, K. (1987). Thermodynamics and control of biological free energy transduction. Amsterdam: Elsevier.

Whitfield, M. L., George, L. K., Grant, G. D., & Perou, C. M. (2006). Common markers of proliferation. *Nature Reviews. Cancer*, *6*(2), 99. doi:10.1038/nrc1802

Wigner, E. (1958). On the distribution of the roots of certain symmetric matrices. *The Annals of Mathematics*, *67*, 325–328. doi:10.2307/1970008

Willadsen, K., & Wiles, J. (2003). Dynamics of gene expression in an artificial genome. In *The 2003 Congress on Evolutionary Computation, 1,* 185-190. IEEE-Press.

Wille, A., & Buhlmann, P. (2006). Low-order conditional independence graphs for inferring genetic networks. *Stat. Appl. Genet. Mol. Biol., 5*(1), Article 1.

Wille, A., Zimmermann, P., Vranova, E., Furholz, A., Laule, O., & Bleuler, S. (2004). Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. *Genome Biology, 5*(11), R92. doi:10.1186/gb-2004-5-11-r92

Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., & Tzur, D. (2007). DrugBank: A knowledgebase for drugs, drug actions, and drug targets. *Nucleic Acids Research, 36*, D901–D906. doi:10.1093/nar/gkm958

Wittkopp, P. J. (2005). Genomic sources of regulatory variation in cis and in trans. *Cellular and Molecular Life Sciences, 62*(16), 1779–1783. doi:10.1007/s00018-005-5064-9

Wolf, D. M., & Arkin, A. P. (2002). Fifteen minutes of *fim*: Control of type 1 pili expression in *E. coli*. *OMICS: A Journal of Integrative Biology, 6*(1), 91–114. doi:10.1089/15362310252780852

Wolf, D. M., & Eeckman, F. H. (1998). The relationship between genomic regulatory element organization and gene regulatory dynamics. *Journal of Theoretical Biology, 195*, 167. doi:10.1006/jtbi.1998.0790

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics, 55*, 601. doi:10.1103/RevModPhys.55.601

Woolf, P. J., & Wang, Y. (2000). A fuzzy logic approach to analyzing gene expression data. *Physiological Genomics, 3*, 9–15.

Wray, B. (1991). Theory refinement on Bayesian networks. *The Seventh Conference on Uncertainty in Artificial Intelligence* (pp. 52-60). Los Angeles: Morgan Kaufmann Publishers Inc.

Wray, G. A. (2003). Transcriptional regulation and the evolution of development. *The International Journal of Developmental Biology, 47*(7-8), 675–684.

Wray, G. A., Hahn, M. W., Abouheif, E., Balhoff, J. P., Pizer, M., Rockman, V., & Romano, L. A. (2003). The evolution of transcriptional regulation in eukaryotes. *Molecular Biology and Evolution, 20,* 1377–1419. doi:10.1093/molbev/msg140

Wu, J., & Weiss, B. (1991). Two divergently transcribed genes, soxr and soxs, control a superoxide response regulon of escherichia coli. *Journal of Bacteriology, 173*(9), 2864–2871.

Wu, W. S., Li, W. H., & Chen, B. S. (2006). Computational reconstruction of transcriptional regulatory modules of the yeast cell cycle. *BMC Bioinformatics, 7,* 421. doi:10.1186/1471-2105-7-421

Xenarios, I., Rice, D. W., Salwinski, L., Baron, M. K., Marcotte, E. M., & Eisenberg, D. (2000). DIP: The database of interacting proteins. [from http://dip.doe-mbi.ucla.edu/]. *Nucleic Acids Research, 28*, 289–291. Retrieved in December 2007. doi:10.1093/nar/28.1.289

Xiong, M., Li, J., & Fang, X. (2004). Identification of genetic networks. *Genetics, 166*(2), 1037–1052. doi:10.1534/genetics.166.2.1037

Xu, H., Wu, P., Wu, C. F., Tidwell, C., & Wang, Y. (2002). A smooth response surface algorithm for constructing a gene regulatory network. *Physiological Genomics, 11,* 11–20.

Yagil, G., & Yagil, E. (1971). On the relation between effector concentration and the rate of induced enzyme synthesis. *J. Biophys., 11*(1), 11–27. doi:10.1016/S0006-3495(71)86192-1

Yeung, M., Tegner, J., & Collins, J. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America, 99*(9), 6163–6168. doi:10.1073/pnas.092576199

Yi, N., George, V., & Allison, D. B. (2003). Stochastic search variable selection for identifying multiple quantitative trait loci. *Genetics, 164*(3), 1129–1138.

Yoo, C., Thorsson, V., & Cooper, G. F. (2002). Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. *Pacific Symposium on Biocomputing, 7,* 498–509.

Yook, S.-H., & Meyer-Ortmanns, H. (2006). Synchronization of Rössler oscillators on scale-free topologies. *Physica A, 371,* 781–789. doi:10.1016/j.physa.2006.04.116

Yu, H., & Gerstein, M. (2006). Genomic analysis of the hierarchical structure of regulatory networks. *Pro-*

ceedings of the National Academy of Sciences of the United States of America, *103*, 14724–14731. doi:10.1073/pnas.0508637103

Yu, H., Kim, P. M., Sprecher, E., Trifonov, V., & Gerstein, M. (2007). The importance of bottlenecks in protein networks: Correlation with gene essentiality and expression dynamics. *PLoS Computational Biology*, *3*(4), e59. doi:10.1371/journal.pcbi.0030059

Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J., & Jarvis, E. D. (2004). Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics (Oxford, England)*, *20*, 3594–3603. doi:10.1093/bioinformatics/bth448

Yu, J., Xiao, J., Ren, X., Lao, K., & Xie, S. (2006). Probing gene expression in live cells, one protein molecule at a time. *Science*, *311*, 1600–1603. doi:10.1126/science.1119623

Yuh, C., Bolouri, H., & Davidson, E. (1998). Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene. *Science*, *279*, 1896–1902. doi:10.1126/science.279.5358.1896

Yvert, G., Brem, R. B., Whittle, J., Akey, J. M., Foss, E., & Smith, E. N. (2003). Transacting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors. *Nature Genetics*, *35*(1), 57–64. doi:10.1038/ng1222

Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, *37*, 77–84. doi:10.1145/175247.175255

Zadeh, L. A. (1996). Fuzzy logic=computing with words. *IEEE transactions on Fuzzy Systems*, *4*, 103–111. doi:10.1109/91.493904

Zak, D. E., Doyle, F. J., Gonye, G. E., & Schwaber, J. S. (2001) Simulation studies for the identification of genetic networks from cDNA array and regulatory activity data. In *Proc. 2nd Intl. Conf. Systems Biology*, 231-238.

Zak, D. E., Gonye, G. E., Schwaber, J. S., & Doyle, F. J. III. (2003). Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: Insights from an identifiability analysis of an in silico network. *Genome Research*, *13*, 2396–2405. doi:10.1101/gr.1198103

Zeng, Z. (1993). Theoretical basis for separation of multiple linked gene effects in mapping quantitative trait loci. *Proceedings of the National Academy of Sciences of the United States of America*, *90*(23), 10972–10976. doi:10.1073/pnas.90.23.10972

Zeng, Z. B. (1994). Precision mapping of quantitative trait loci. *Genetics*, *136*(4), 1457–1468.

Zeng, Z.-B., Liu, J., Stam, L. F., Kao, C.-H., Mercer, J. M., & Laurie, C. C. (2000). Genetic architecture of a morphological shape difference between two *Drosophila* species. *Genetics*, *154*(1), 299–310.

Zenke, M., & Hieronymus, T. (2006). Towards an understanding of the transcription factor network of dendritic cell development. *Trends in Immunology*, *27*(3), 140–145. doi:10.1016/j.it.2005.12.007

Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, *4*, 17. doi:10.2202/1544-6115.1128

Zhang, D., & Rosbash, M. (1999). Identification of eight proteins that cross-link to preRNA in the yeast commitment complex. *Genes & Development*, *13*(5), 581–592. doi:10.1101/gad.13.5.581

Zhang, S., Jin, G., Zhang, X. S., & Chen, L. (2007). Discovering functions and revealing mechanisms at molecular level from biological networks. *Proteomics*, *7*(16), 2856–2869. doi:10.1002/pmic.200700095

Zhao, W., Serpedin, E., & Dougherty, E. R. (2006). Inferring gene regulatory networks from time series data using the minimum description length principle. *Bioinformatics (Oxford, England)*, *22*(17), 2129–2135. doi:10.1093/bioinformatics/btl364

Zhao, Y., & Karypis, G. (2005). Data clustering in life sciences. *Molecular Biotechnology*, *31*(1), 55–80. doi:10.1385/MB:31:1:055

Zhi-fen, Z., Tong-ren, D., Wen-zao, H., & Zhen-xi, D. (1992). *Qualitative theory of differential equations.* American Mathematical Society.

Zhou, L., Mideros, S. X., Bao, L., Tripathy, S., Torto-Alalibo, T. A., Mao, Y., et al. (2008). *Dissecting soybean resistance to Phytophthora by QTL analysis of host and pathogen expression profiles.* Paper presented at the International Plant and Animal Genome Conference XVI, San Diego.

Zhou, X., Wang, X., Pal, R., Ivanov, I., Bitner, M., & Dougherty, E. R. (2004). A Bayesian connectivity-based approach

to constructing probabilistic gene regulatory networks. *Bioinformatics (Oxford, England), 20*(17), 2918–2927. doi:10.1093/bioinformatics/bth318

Zhou, X., Wang, X., Pal, R., Ivanov, I., Bittner, M. L., & Dougherty, E. R. (2004). A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics (Oxford, England), 20*(17), 2918–2927. doi:10.1093/bioinformatics/bth318

Zhu, J., Jambhekar, A., Sarver, A., & DeRisi, J. (2006). A Bayesian network driven approach to model the transcriptional respone to nitric oxide in Saccharomyces cerevisiae. *PLoS ONE, 1*(1), e94. doi:10.1371/journal.pone.0000094

Zhu, J., Lum, P. Y., Lamb, J., GuhaThakurta, D., Edwards, S. W., & Thieringer, R. (2004). An integrative genomics approach to the reconstruction of gene networks in segregating populations. *Cytogenetic and Genome Research*, *105*(2-4), 363–374. doi:10.1159/000078209

Zhu, J., Wiener, M. C., Zhang, C., Fridman, A., Minch, E., & Lum, P. Y. (2007). Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. *PLoS Computational Biology, 3*(4), e69. doi:10.1371/journal.pcbi.0030069

Zhu, R., Ribeiro, A. S., Salahub, D., & Kauffman, S. A. (2007). Studying genetic regulatory networks at the molecular level: Delayed reaction stochastic models. *Journal of Theoretical Biology, 246*(4), 725–745. doi:10.1016/j.jtbi.2007.01.021

Zhu, X., Yin, L., Hood, L., Galas, D., & Ao, P. (2007). Efficiency, robustness, and stochasticity of gene regulatory networks in systems biology: λ switch as a working example. In S. Choi (Ed.), *Introduction to systems biology*. Humana Press.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B. Methodological, 67*(2), 301–320. doi:10.1111/j.1467-9868.2005.00503.x

Zou, M., & Conzen, S. D. (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics (Oxford, England), 21*(1), 71–79. doi:10.1093/bioinformatics/bth463

About the Contributors

Sanjoy Das is an associate professor in the Department of Electrical & Computer Engineering at Kansas State University. He received a Ph.D. in Electrical Engineering from Louisiana State University in 1994. He was a postdoctoral researcher at the University of California, Berkeley and the Smith-Kettlewell Institute between 1994 and 1997. Between Until 2001 he held various research appointments in the industry. Prof. Das's research interests include computational intelligence, bio-inspired computing, and their applications to genomics (especially gene regulatory network modeling). He has published over 100 research papers in journals, books and conference proceedings. His research has been funded by the U.S. National Science Foundation, the U.S. Department of Agriculture, and the U.S. Department of Defense.

Doina Caragea is an assistant professor at Kansas State University. Her research interests include artificial intelligence, machine learning, data mining, information integration and information visualization, with applications to bioinformatics. Doina received her Ph.D. in Computer Science from Iowa State University in August 2004 and was honored with the Iowa State University Research Excellence Award for her achievements. Her Ph.D. work at Iowa State University was focused on learning classifiers from autonomous, distributed, semantically heterogeneous data sources. Her recent work at Kansas State University has been focused on the development of algorithms and tools for genome annotation. More specifically, she has participated in projects such as EST data analysis, investigation of transcription networks and their relation to environment, and studies on alternative splicing, among others. Prof. Caragea has published more than 30 refereed conference and journal articles. She is teaching machine learning, data mining and bioinformatics courses.

Stephen M. Welch is a professor at Kansas State University. His focus is gene networks, plant phenology, optimal parameter estimation, and parallel computing, with applications in ecological genomics and plant breeding. He has a B.S. in Computer Science (1971) and a Ph.D. in Zoology (1977), both from Michigan State University. The common thread in his career has been computer simulation of living systems in both the departments of entomology and (since 1990) agronomy. Short term activities have included service as Acting State Climatologist for Kansas and Interim Director of University Computing and Network Services. Recent work has involved modeling the genetic control of Arabidopsis flowering time as part of a multinational collaboration with field sites from Spain to Finland. Under the auspices of the iPlant Collaborative funded by the US National Science Foundation, he also co-leads an international team developing a cyberinfrastructure for grand challenge research that interrelates plant genotypes and phenotypes. He has 61 peer reviewed papers, conference proceedings, and book chapters, plus 78 publications of other types.

William H. Hsu is an associate professor in the Department of Computing and Information Sciences at Kansas State University. He received a B.S. in Mathematical Sciences and Computer Science and an M.S.Eng. in Computer Science from Johns Hopkins University in 1993, and a Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign in 1998. His research interests include machine learning and probabilistic reasoning, with applications to information extraction, time series prediction, link mining, and bioinformatics (especially computational genomics and proteomics).

* * *

Ugo Ala graduated in physics at the Università di Torino, Italy, with a research thesis on quantum mechanics entitled "Numerical study of Wigner function and applications." After some years as teacher of Mathematics and Physics in intermediate schools when he collaborated at the writing of a physics text-book, he obtained a master's degree in Bioinformatics from the same university joined with the "Biotechnology foundation of Turin" with the thesis entitled "Implementation of a tool for human-mouse co-expression analysis based on Affymetrix data." He became a Ph.D. student in Molecular Biotechnology, with focus on Bioinformatics, at the Molecular Biotechnology Center and the Department of Genetics, Biology and Biochemistry of the Università di Torino. Joining the activity of tutor in practical courses of statistics and programming, he attending to several congress also in quality of speaker. His main scientific topics are gene co-expression studies and phylogenetic analysis.

Mr. **Almasri** received a Mechanical Engineering degree in 1999 from Jordan University of Science and Technology in Jordan. He received his M.S. degrees in Mechanical and Aerospace Engineering from Illinois Institute of Technology, Chicago in 2003. Currently, he is a PhD candidate in Bioinformatics at the University of Illinois at Chicago where he works as a doctoral research assistant in the Dai Laboratory. During his PhD study, he designed and implemented probabilistic models to predict biological networks. In particular, he developed a Bayesian network framework for the prediction of gene regulatory networks using prior biological knowledge from literature publications. He also developed a probabilistic model to predict protein-protein interactions using Gene Ontology. His research interests are focused on functional genomics, Bayesian network, classifications and mathematical modeling of biological networks.

Manuel Barrio is an Associate Professor in the Department of Computer Science at the University of Valladolid, Spain. He received his M.Sc. in Physics and his Ph.D. in Computer Science from the University of Valladolid. His research interests are in computational models and simulation algorithms applied to biological systems. Specifically, his research focuses on stochastic modelling for gene regulatory networks and the role of delays and intrinsic noise in cellular regulation.

Sebastian Bauer is a computer scientist who graduated from the Ilmenau University of Technology located in Ilmenau, Germany. Currently he is doing his PhD at the Institute for Medical Genetics of the Charité Universitätsmedizin Berlin. His research interests span the wide area of discrete algorithms, which includes graph algorithms and optimization problems, and their application in medicine and biology. He is interested in distributed and parallel computing. He is furthermore involved in the process of modeling and inferring biological networks. Here his main focus is the integration of experimental data gained by variety of different sources, for instance by microarray assays with deterministic and stochastic mathematical models.

Daniel Bryce is an assistant professor of computer science at Utah State University in Logan, UT, since 2008, where he conducts research on artificial intelligence and systems biology. Bryce received his Ph.D. in computer science from Arizona State University in Tempe, AZ, in 2007. Prior to joining Utah State University, Bryce was a computer scientist at SRI International in Menlo Park, CA and a visiting lecturer at Stanford University. Bryce's ongoing research projects address instructable computing, planning under uncertainty, and applications of planning to systems biology.

Svetlana Bulashevska graduated in mathematics at the University of Odessa, Ukraine, and in computer science at the Elite University of Karlsruhe, Germany. Since 2000 she is working at the Department "Theoretical Bioinformatics" at the German Cancer Research Centre (DKFZ) in Heidelberg. She received her PhD degree in bioinformatics in 2004. Her main research field is model-based analysis of biological systems, with focus on developing mathematical and statistical approaches to address problems of bioinformatics and systems biology. Her research interests include modelling genetic regulation, modelling and simulation of signal transduction, statistical modelling of large biological networks, computational oncology.

Professor **Burrage** is currently Professor of Computational Systems Biology at the University of Oxford and Professorial Research Fellow at the IMB, University of Queensland. He has done fundamental algorithmic and mathematical modelling work in Computational Mathematics (numerical solution of Ordinary Differential and Stochastic Differential Equations, and numerical algorithms for linear systems); advanced computing (parallel and grid computing); modelling (financial mathematics, environmental modelling and computational engineering) and most recently in Computational Systems Biology (with a focus on the role of noise in Biological systems) through his Federation Fellowship awarded by the Australian Research Council between 2003-2008. He has over 180 scientific publications.

Dr **Pamela Burrage** is a graduate from the University of Auckland, New Zealand. She is currently a research fellow in Computational Systems Biology at the Institute for Molecular Bioscience, University of Queensland. She has done fundamental algorithmic and mathematical modelling work in Computational Mathematics (numerical solution of Ordinary Differential and Stochastic Differential Equations; advanced computing (parallel and grid computing) and most recently in Computational Systems Biology. Her current research project is the modelling and simulation of chemical kinetics on the Plasma Membrane of a cell using grid computing. She has over 20 scientific publications.

Mr. **Guanrao Chen** was born in Chengdu, China. He received his B.E. in engineering mechanics from Chongqing University, China in 1997. Fascinated with computer technology, he switched his academic interest and obtained his M.E. in computer engineering from Sichuan University, China in 2000. His specialty then was image processing, especially with medical images. To pursue advanced education, he came to the United States and finished his Ph.D. of computer science in the University of Illinois at Chicago in 2008. His research is mainly on bioinformatics with the topics focused on identification and prediction of biological networks. He has several papers published on journals and in conferences such as BMC Bioinformatics and EMBS.

Luonan Chen is a Professor at Department of Electrical Engineering and Electronics, Osaka Sangyo University, and is also with ERATO Aihara Complexity Modeling Project of JST, Institute of Industrial

Science of The University of Tokyo, Japan, and Institute of Systems Biology, Shanghai University, China. He received his Bachelor degree from Department of Electrical Engineering, Huazhong University of Science and Technology (Wuhan, China) in 1984, He received his Master degree (1988) and Ph.D. degree (1991) from Department of Electrical and Communication Engineering, Tohoku University (Sendai, Japan). He is IEEE Senior member and the member of IEE Japan. His current interests are in systems biology and bioinformatics.

Dr **Madhu Chetty** has been working at Monash University, Australia for over thirteen years. His main research interests include but not limited to the application of Computational Intelligence (CI) techniques such as Neural Network, Genetic Algorithm, Bayesian Network, Fuzzy system etc to problems from bioinformatics. He has supervised number of students for their doctoral work in bioinformatics and has number of publications in journals and international conferences. He is also recipient of number of grants to support his research. He is the senior member of IEEE and has also served as vice chair of the IEEE technical committee on bioinformatics and bioengineering. Dr Chetty was the General Chair of the recently concluded PRIB'08 (Pattern Recognition in Bioinformatics) conference at Melbourne and is currently serving as associate editor of the Elsevier's Neurocomputing journal and is on the editorial board of few journals in bioinformatics.

Tianjiao Chu is an assistant professor at University of Pittsburgh School of Medicine. In 2003 he got his PhD degree in logic, computation, and methodology from Carnegie Mellon University. His main research focus is in the field of bioinformatics. He is also interested in causal discovery and its applications. He has published papers on topics including machine learning, bioinformatics, and genetics.

Adriana Climescu-Haulica received a Ph.D. in mathematics from Ecole Polytechnique Fédérale de Lausanne in 1999 and a M.S. in computer science from the University of Iasi, Romania in 1990. As a research scientist with Communication Research Centre in Ottawa she worked on stochastic calculus applied to random signal detection. During her lecturer appointment with the Institut National Polytechnique de Grenoble she begun her research in computational biology. She was a scientist with Laboratoire Information Génomique et Stucturale - Conseil National de la Recherche Scientifique and Laboratoire Biologie-Informatique-Mathématiques, Commisariat Energie Atomique, France. Her research interests include system biology, probabilistic learning, and statistical data mining. Her work addressed the mathematical modelling of transcriptional regulatory networks, biomarkers discovery, and the study of noncodingRNAs. She is currently the research lead of the Computational Biomedicine Investigation Project, in partnership with the Laboratoire Techniques de l´Ingénierie Médicale et de la Complexité-Informatique-Mathématiques Appliquées de Grenoble.

Ernesto Costa is Full Professor at the Department of Informatics Engineering of the University of Coimbra, where he concluded its B.Sc. in 1976. He received a 3rd Cycle Thesis in Computing Science from the University Pierre et Marie Curie (Paris, France) in 1981 and got a Ph.D. in Electronic Engineering (area of Computing Science) from the University of Coimbra (Portugal) in 1985. His current research interests are in the areas of Evolutionary Computation, Complex Systems, and Computational Biology. He was the founder and Head of the Evolutionary and Complex Systems Group of the Centre for Informatics and Systems of the University of Coimbra. He participated in several projects, orga-

nized several international scientific events and had published over 150 papers in books, journals and proceedings of conferences. He was the recipient of three international prizes.

Yang Dai received Ph.D degree in Management Science and Engineering from University of Tsukuba, Japan, in 1991. From 1991 to 1997 she was a research associate, an assistant professor at the Department of Management Science of Kobe University of Commerce. From 1997 -2001 she was an assistant professor of Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan. Since 2001 she is on faculty of Department of Bioengineering, the University of Illinois at Chicago. Her research concerns the development of efficient algorithms for discrete and continuous optimization problems arising in the fields of computer science, engineering, and biology. Professor Dai recently has focused on bioinformatics problems related to functional genomics such as the analyses of microarray gene expression and pathways, prediction of protein-protein interactions, immunoinformatics, and genome-wide association study of complex diseases. Professor Dai has published more than 60 peer reviewed papers in various journals and conferences proceedings.

Christian Damasco is a Ph.D. student at the Università di Torino from January 2008. He is following a Ph.D. course in Molecular Biotechnology and their main projects are focused on bioinformatic approaches to explore genes coexpression in Drosophila Melanogaster and mammals. He graduated in July 2006 in Molecular Biotechnology at the Università di Torino with a research thesis on bioinformatics methods based on genes coexpression analysis to discovery new mitotic genes. From January 2007, he collaborates as teaching assistant for the degree course in Biotechnology at the University of Turin and for the master's degree course in Bioinformatics promoted by the Biotechnology Foundation of Turin.

David Danks is an Associate Professor of Philosophy and Psychology at Carnegie Mellon University, as well as a Research Scientist at the Institute for Human & Machine Cognition. He received a Ph.D. in philosophy from the University of California, San Diego in 2001, and came to CMU in 2003. Dr. Danks's principal research interests are in computational cognitive science and machine learning, with particular interests in causal learning and reasoning, concept learning, and categorization. He has published in journals in computer science, psychology, and philosophy.

Christian Darabos graduated in 2004 from the University of Lausanne, Switzerland. He obtained a Master's Degree in Computer Science, and specialized in bio-inspired computation and optimization methods, such as Genetic Algorithm and Genetic Programming. After a brief experience in the industry, he returned to the University of Lausanne to undertake a Ph.D. in collaboration with the Molecular Biotechnology Center of the University of Turin, Italy. He is currently in the final year of the 5 year joint doctoral program between the Information Systems Department, under the supervision of Prof. Marco Tomassini in Lausanne, and Prof. Ferdinando di Cunto in Turin. He is also employed as a teaching and research assistant. He focuses his efforts in Complex Systems, particularly the dynamics of Biological Complex System. He is expected to graduate later in 2009.

Ana Teresa Freitas is an Assistant Professor at the Department of Electrical and Computer Engineering of Instituto Superior Técnico. She is also a senior researcher at INESC-ID in the Knowledge Discovery and Bioinformatics group (KDBIO). She got the B.S. and Ms.C. degree in Electrical and Computer Engineer in 1990 and 1994, respectively, from IST, and a PhD in the area of Computer Aided

Design with applications to dynamic systems modeling in 2002. She has co-authored more than 30 papers in journals and international conferences in the areas of computational biology, bioinformatics and computer aided design. Her research interests are now centered in the areas of computational biology and bioinformatics, data mining, algorithms and complexity.

Dr. **Alberto de la Fuente** obtained his PhD degree at the Free University Amsterdam, while carrying out his research at the Virginia Bioinformatics Institute (VBI) with the main focus on Gene Regulatory Network inference. After graduating he stayed at VBI for a Post Doc position focusing on the integration of statistical genetics with network inference approaches. He currently is a Senior Researcher at CRS4 Bioinformatica in Pula, Sardinia, Italy. His main interests are reverse-engineering bio-molecular networks, studying their topology and dynamics, and to investigate these networks in the context of different 'omics' data sources. The overall goal of his work is to contribute to our understanding of complex diseases through the identification and analysis of bio-molecular networks.

Mario Giacobini has received a PhD in Computer Science from the University of Lausanne (Switzerland) and the University of Milano (Italy), with a work on Evolutionary Algorithms and Artificial Life. He is now Researcher at the Faculty of Veterinary Medicine of the University of Torino, pursuing his researches at the Department of Animal Production, Epidemiology and Ecology, and at the Molecular Biotechnology Center. His main research interests range from Computational Biology to Artificial Life, and from Evolutionary Game Theory to Computational Epidemiology. In particular, he is interested in how the communication topology of the particles that interact in a system influence its dynamics.

Dr. **John Grefenstette** is a Professor of Bioinformatics and Computational Biology at George Mason University. He obtained his B.S. in Mathematics from Carnegie Mellon University and his M.S. and Ph.D. in Computer Science from the University of Pittsburgh. He previously served as Head of the Machine Learning at the Naval Research Laboratory in Washington, DC. He also served as Chair of the Bioinformatics and Computational Biology Program at George Mason. Dr. Grefenstette serves on the editorial board for the journal Adaptive Behavior and has been Associate Editor for the journals Evolutionary Computation and Machine Learning. His research interests include machine learning, evolutionary algorithms, computational models of biological networks, and bioinformatics.

Clark Glymour received Bachelor's degrees in Chemistry and Philosophy from the University of New Mexico, and a Ph.D. in History and Philosophy of Science from Indiana University in 1969. He has been a Guggenheim Fellow, a Fellow of the Center for the Advanced Study in the Behavioral Sciences, a Phi Beta Kappa Romanelli Fellow, and is a Fellow the Statistics Section of the American Association for the Advancement of Science. His books include Theory and Evidence (Princeton, 1980); Foundations of Space-Time Physics (Minnesota, 1981); Examining Holistic Medicine (Prometheus, 1983); Discovering Causal Structure (Academic, 1987); Causation, Prediction and Search (Springer, 1993; MIT, 2001); Thinking Things Through (MIT, 1997); Android Epistemology (MIT, 2001); and The Mind's Arrows (MIT, 2003).

Ângela Gonçalves concluded, in 2007, a M.Sc. in Informatics Engineering at the University of Coimbra, Portugal, with the thesis "A Computational Model for Genetic Regulatory Networks" under the supervision of Professor Ernesto Costa. Also in the University of Coimbra she was tutor of Theory

of Computation, Artificial Intelligence and Programming classes. From 2007 to 2008 she was a Trainee at the European Space Agency's Centre for Earth Observation in Rome where she worked in the development of software applications for the visualization and processing of earth observation data and in service chain modelling. She is, since 2008, a Ph.D. student at the University of Cambridge and the European Bioinformatics Institute.

Mika Gustafsson received a M.Sc. in mathematics from Stockholm University in 2003 and a M.Sc. in physics from Linköping University, Sweden, in the same year. Now he works on a Ph.D. dissertation in applied mathematical physics at Linköping Institute of Technology, focusing mainly on inference and analysis of networks.

Dr. **Hoeschele** obtained her PhD at Hohenheim University, Stuttgart, Germany, followed by postdoctoral work at Iowa State University and at the University of Illinois at Champaign-Urbana, USA. She is a Professor of Statistics at the Virginia Bioinformatics Institute and in the Department of Statistics at Virginia Tech, Blacksburg (VA), as well as an Adjunct Professor at the Wake Forest University Medical School in Winston-Salem (NC), USA. Dr. Hoeschele is a statistical geneticist with current main research interests in highly multivariate, Bayesian parametric and nonparametric, multiple quantitative trait loci linkage and association mapping, in causal network inference from systems genetics experiments, and in basic statistical analyses of 'omics' data. The overall goal of her work is to contribute to our understanding of the genetic basis of complex diseases.

Michael Hörnquist is an associate professor of theoretical physics at Linköping University, Sweden. He received a Ph.D. from Linköping Institute of Technology in 1999. His research comprises various aspects of theoretical biological physics, such as computational systems biology and DNA dynamics.

Dr. **Marc-Thorsten Hütt** studied physics in Göttingen and Paris. He received is Ph.D. in 1997 from Göttingen University. After research stays in Hamburg, Warsaw, Novosibirsk and Helsinki he joined Darmstadt University of Technology as a postdoc (1998) and later as an Assistant Professor of Theoretical Biology and Bioinformatics (2002). Since 2006 he is Professor of Computational Systems Biology at Jacobs University in Bremen, Germany. From 2000 to 2005 he was a member of Die Junge Akademie (a joint institution of Berlin-Brandenburger Akademie der Wissenschaften and Deutsche Akademie der Naturforscher Leopoldina). His research interests include spatiotemporal dynamics and pattern formation phenomena in biology, network dynamics and properties of biological networks, as well as the theory of self-organization and its applications to biology. His books ("Datenanalyse in der Biologie", Springer 2001 and, together with Manuel Dehnert, "Methoden der Bioinformatik", Springer 2006) provide bridges between theory and experiment in an attempt to understand biological systems on many different scales.

Ivan Ivanov received BS and MS degrees in Mathematics from the Department of Mathematics, Sofia University, Bulgaria, and PhD in Mathematics at University of South Florida. He did postdoctoral work in Mathematics at Syracuse University and Texas A&M University, and was a postdoctoral trainee in the Training Program in Bioinformatics, Texas A&M University. He has been with the Department of Veterinary Physiology and Pharmacology, Texas A&M University since 2006, where he is currently an Assistant Professor. He also holds a joint appointment at the Department of Computer and Electrical

Engineering, Texas A&M University and collaborates actively with the members of Genomic Signal Processing laboratory, Texas A&M University. For the past five years his research has been primarily in the field of Genomic Signal Processing with an emphasis on mathematical modeling of genomic regulatory networks.

Yaochu Jin received the Ph.D. degree in Automatic Control from Zhejiang University, China in 1995 and the Dr.-Ing. degree in Neuroinformatics from Ruhr-University Bochum, Germany in 2001. Presently, he is a Principal Scientist at the Honda Research Institute Europe, Offenbach, and Scientific Coordinator, CoR-Lab Graduate School, Bielefeld University, Bielefeld. His research interests include artificial life, systems biology, and computational intelligence. Dr. Jin currently serve as an Associate Editor of the IEEE Computational Intelligence Magazine, IEEE Transactions on Neural Networks, and the IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews. He has been a Program (Co)-Chair and an invited Plenary Speaker of several international confereces and symposiums. He is a Senior Memeber of IEEE.

Dr. **Viktor Jirsa** earned a Ph.D. degree in Theoretical Physics from the University of Stuttgart. He is currently Director of Research at the Centre National de la Recherche Scientifique in Marseille, France, and Associate Professor at Florida Atlantic University in the Center for Complex Systems and Brain Sciences and the Department of Physics. His research focusses on the understanding of the mechanisms underlying the emargence of low-dimensional (often cognitive) behavior from high-dimensional network dynamics.

Trupti Joshi is a Research Associate and Lab Manager in the Digital Biology Laboratory, Computer Science Department at the University of Missouri-Columbia. She earned an MS Degree with computational biology and bioinformatics major from the University of Tennessee-Oak Ridge National Laboratory, Graduate School of Genome Science and Technology. Her research interests are in the areas of data mining, analysis of high-throughput biological data for function and biological pathway prediction, regulatory networks identification, SNP discovery using new generation sequencing technologies including 454 and Illumina.

Lars Kaderali received a PhD in Computer Science from the University of Cologne. He worked as postdoc in the department of Theoretical Bioinformatics at the German Cancer Research Center in Heidelberg, and is now department head in Computational Systems Biology at the University of Heidelberg. His research focuses on the application of machine learning and statistical approaches in Bioinformatics and Systems Biology, on the reconstruction of signal transduction and gene regulatory networks from high-throughput experimental data, and on mathematical modeling of virus-host interactions. Dr. Kaderali has published over a dozen original research papers, one book and several contributed book-chapters. He is member of the excellence cluster cellular networks at the University of Heidelberg, and faculty member of the Hartmut-Hoffmann-Berling International Graduate School of Molecular and Cellular Biology.

Stuart A. Kauffman, is a professor at the University of Calgary with a shared appointment between biological sciences and physics and astronomy. He is also the leader of the Institute for Biocomplexity and Informatics (IBI) which conducts leading-edge interdisciplinary research in systems biology. Dr.

Kauffman is also an emeritus professor of biochemistry at the University of Pennsylvania, a MacArthur Fellow and an external professor at the Santa Fe Institute. Originally a medical doctor, Dr. Kauffman's primary work has been as a theoretical biologist studying the origin of life and molecular organization. Thirty-five years ago, he developed the Kauffman models, which are random networks exhibiting a kind of self-organization that he terms "order for free". Dr. Kauffman is the author of The Origins of Order, At Home in the Universe: The Search for the Laws of Self-Organization, Investigations and Reinventing the Sacred: A New View of Science, Reason, and Religion.

Seungchan Kim, PhD, was trained in Math and Electrical Engineering at Texas A&M University, and in Computational Biology at the Cancer Genetics branch at NIH. As a founding member of the Translational Genomics Research Institute (http://www.tgen.org), he is jointly appointed as an Assistant Professor in Computer Science and Engineering at Arizona State University, which uniquely positions Dr. Kim to collaborate with leading computer and biomedical scientists. His research focuses on mathematical modeling and inference of gene regulatory networks. In his previous works, Dr. Kim has developed several discrete-valued and continuous-valued mathematical models for gene regulatory networks, including Probabilistic Boolean networks (PBNs) that allow stochasticity in Boolean networks. Recently, he is studying contextual specificity of genomic regulation, especially in cancer development.

Ina Koch studied theoretical chemistry at University of Leipzig with specialization in quantum chemistry. After study she worked on protein structure analysis and prediction at Central Institute for Cybernetics and Information Processes at Academy of Sciences. Ina Koch received her Dr. rer. nat. (PhD) in computer science. The PhD thesis considered the development of a fast algorithm for protein structure alignment. Since about ten years Ina Koch is interested in systems biology, in particular in Petri net modeling techniques applied to biochemical systems. She has experience in modeling metabolic systems, signal transduction networks, and gene regulatory networks. She occupied a temporary bioinformatics professorship at University of Jena and is now a professor for bioinformatics at Technical University of Applied Sciences Berlin doing her research at the Max Planck Institute for Molecular Genetics Berlin. Her research interest is method development for bioinformatics and computational systems biology.

Yasuaki Kuroe received his Ph.D. in industrial science from Kobe University, Kobe, Japan in 1982. In the same year he joined the faculty of Department of Electrical Engineering, Kobe University as a Research Associate. In 1991, he moved to the Department of Electronics and Information Science, Kyoto Institute of Technology as an Associate Professor. He is currently a Professor at the Department of Information Science, Kyoto Institute of Technology, Kyoto, Japan. In 1996, he was a visiting Research Scientist at the Massachusetts Institute of Technology. His research interests are in the areas of neurocomputing and computational intelligence, control theory and its application, and computer-aided analysis and design.

Hiroyuki Kuwahara received the B.S. and Ph.D. degrees in Computer Science from the University of Utah, Salt Lake City, UT, U.S.A. in 2001 and 2008, respectively. He currently holds a junior researcher position at The Microsoft Research - University of Trento Centre for Computational and Systems Biology, Trento, Italy. His research interests include development of computational methodologies to analyze dynamics of biochemical networks and computational analysis of systems-level properties in biochemical networks to offer new insights.

Mr. **Peter Larsen** received his B.S. degree in Biological Sciences from Purdue University in 1993 and his B.S. in Bioengineering from University of Illinois at Chicago. Mr. Larsen has nearly fifteen years biotechnology research experience from Chicago area biotechnology companies and academic laboratories and has worked both at the bench and in computational analysis. His research background includes engineering thermostable proteins for industrial purposes, metabolic engineering of antibiotic-producing bacteria, and gene expression and aCGH microarray analysis. His current research interests are in making use of the many available large databases of biological facts in the prediction of biologically relevant interaction networks. He has a number of scientific publications and a patent.

Dmitriy Laschov was born in Russia in 1973. He received his B.Sc. degree in Electrical Engineering (EE) from the East Ukraine National University, in 1999, and his M.Sc. in EE from the School of Electrical Engineering –Systems, Tel-Aviv University, in 2008. He is currently employed as a software engineer in RAD Data Communications, Israel.

André Leier is a postdoctoral research fellow in the Department of Biosystems Science and Engineering, Swiss Federal Institute of Technology Zurich, previously at the Advanced Computational Modelling Centre, University of Queensland, Australia. He received his PhD in Computer Science from the University of Dortmund, Germany. His research interests include Computational and Systems Biology, Synthetic Biology, and Bioengineering, in particular, stochastic, spatio-temporal multi-scale modeling of cell signaling and genetic regulation and the role of delays in cellular processes.

Dr. **Larry S. Liebovitch** earned a B.S. in Physics from the City University of New York and a Ph.D. in Astronomy from Harvard University. He was an Assistant Professor at Columbia University, is currently a Professor at Florida Atlantic University in the Center for Complex Systems and Brain Science, Center for Molecular Biology and Biotechnology, Department of Psychology, and the Associate Dean for Graduate Studies and Programs in the Charles E. Schmidt College of Science. He uses fractals, chaos, networks, and other nonlinear methods to study molecular, cellular, physiological, and psychological systems which have provided insights into the structure and motion of ion channel proteins in the cell membrane, the timing of heart attacks, the spread of electronic and biological infections, the spatial pattern of artifacts found in archeological sites, the network of gene regulation, and the dynamics of conflicts between people.

Dr. Bing Liu obtained her PhD degree in Statistics under the instruction of Dr. Ina Hoeschele at the Virginia Bioinformatics Institute and the Department of Statistics at Virginia Tech, Blacksburg (VA). Dr. Liu's PhD research was mainly focused on causal gene network inference via structural equation modeling. After graduation she joined the Monsanto Co, the world's largest seed company. She is a statistical geneticist working for Monsanto's global molecular breeding and crop discovery programs. The overall goal of her research is to contribute to Monsanto's "sustainable yield initiative" – doubling yields in corn, soybeans and cotton by the year 2030, conserving more resources by decreasing inputs by one third per unit of output and improving farmers' lives.

Michael Margaliot received the B.Sc. (cum laude) and M.Sc. degrees in Electrical Engineering from the Technion – Israel Institute of Technology – in 1992 and 1995, respectively, and the Ph.D. degree (summa cum laude) from Tel Aviv University in 1999. He was a post-doctoral fellow in the Department

of Theoretical Mathematics at the Weizmann Institute of Science. In 2000, he joined the faculty of the School of Electrical Engineering-Systems, Tel Aviv University. Dr. Margaliot's research interests include stability analysis of differential inclusions and switched systems, optimal control, fuzzy control, computation with words, knowledge-based neurocomputing, and fuzzy modeling of biological phenomena. He is co-author of New Approaches to Fuzzy Modeling and Control: Design and Analysis (World Scientific, 2000) and of Knowledge-Based Neurocomputing: A Fuzzy Logic Approach, (Springer-Verlag, 2009).

Tatiana T. Marquez-Lago studied her undergraduate degrees in Mexico city (UNAM, UP) later on obtaining her MSc and PhD in Mathematics from Simon Fraser University and the University of New Mexico, respectively. As a postdoctoral fellow, she has performed research for the University of Queensland, Australia and, as of today, for the Swiss Federal Institute of Technology, Zurich. Her research mainly focuses on mathematical and computational techniques for fast and accurate simulations of chemical kinetics and cell-signalling processes. She also performs active research in pure mathematics (Numerical Analysis, Dynamical Systems), Synthetic Biology and Archaeology.

Carsten Marr received his diploma in general physics from Technical University Munich in 2002. He worked on quantum information at Max-Planck-Institute for Quantum Optics in Garching, Germany, and at Imperial College London, UK. He earning a Ph.D. in Biology from Technische Universität Darmstadt in 2007 and afterwards joined Jacobs University as a postdoctoral fellow, focusing on gene regulatory networks. Since January 2008, he is a member of the Computational Modeling in Biology group at Helmholtz Zentrum München, Germany, where he works on models of gene regulation, the integration of non-coding RNAs into regulatory networks and the large-scale analysis of biological networks.

Kenneth L. McMillan received the B.S. degree in electrical engineering from the University of Illinois, Urbana, in 1984, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1986, and the Ph.D. degree in computer science from Carnegie Mellon University, Pittsburgh, PA, in 1992. He has been a Chip Designer, Biomedical Engineer, a Member of the Technical Staff at AT&T Bell Laboratories, and is currently a Research Scientist at Cadence Research Laboratories, in Berkeley, CA. His current research interests include computer music, formal verification, and design methodology.

Chris Myers received the B.S. degree in electrical engineering and Chinese history in 1991 from the California Institute of Technology, Pasadena, CA, and the M.S.E.E. and Ph.D. degrees from Stanford University, Stanford, CA, in 1993 and 1995, respectively. He is a Professor in the Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT. Dr. Myers is the author of over 80 technical papers and the textbook Asynchronous Circuit Design. He is also a co-inventor on 4 patents. His research interests include algorithms for the analysis of real-time concurrent systems, analog error control decoders, formal verification, asynchronous circuit design, and the modeling and analysis of genetic regulatory circuits. Dr. Myers received an NSF Fellowship in 1991, an NSF CAREER award in 1996, and best paper awards at Async1999 and Async2007.

Yoshihiro Mori received his Ph.D. in engineering from Kyoto Institute of Technology, Kyoto, Japan in 2002. In 1995 he joined the faculty of Department of Electronics and Information Science, Kyoto Institute of Technology as a Research Associate. He is currently a Research Associate at the Department

of Information Science, Kyoto Institute of Technology, Kyoto, Japan. His research interests are in the areas of systems biology and control theory and its applications.

Arlindo L. Oliveira received the BSc and MSc degrees in electrical and computer engineering from Lisbon Technical University, and the PhD degree in electrical engineering and computer science from the University of California, Berkeley, in 1986, 1989 and 1994, respectively. He is currently a professor at Instituto Superior Técnico, Lisbon Technical University. He is also a senior researcher at the Knowledge Discovery and Bioinformatics (KDBIO) group of INESC-ID. His research interests include bioinformatics, systems biology, string processing, algorithm design, combinatorial optimization, machine learning, logic synthesis and automata theory.

Helio Pais his MSc in Computer Science and Engineering degree from Instituto Superior Tecnico, Technical University of Lisbon, Portugal, in 2006. He has been a member of the Knowledge Discovery and Bioinformatics (KDBIO) group of INESC-ID, Lisbon, Portugal, and an intern at Cadence Research Laboratories, Berkeley, California, USA. He is a student of the PhD Program in Computational Biology organised by Instituto Gulbenkian de Ciência and is currently conducting his research training at the University of East Anglia, Norwich, United Kingdom, within the topic 'Computational Biology of microRNAs'.

Michelle D. Quirk. is a scientist at Los Alamos National Laboratory. She holds a Ph.D. in mathematics and a M.S. in computational sciences from the University of Texas in Austin, and a B.S. in theoretical mechanics from the University of Bucharest. Her expertise spans across computational physics, image processing, fuzzy logic and soft computing, systems analysis, and decision analysis under uncertainty. Her research interests focus on suited mathematical and computational paradigms for building effective decision-making tools for complex systems and hybrid networks. In particular, she addresses problems of gene regulatory networks aimed to devise reliable reverse engineering methods.

Nicole Radde received her PhD in Applied Mathematics from the University of Cologne and worked as a postdoctoral researcher at the Institute for Medical Informatics, Statistics and Epidemiology at the University of Leipzig thereafter. Since October 2008, she has an appointment as a Junior Professor 'Systems Theory in Systems Biology' at the Institute for Systems Theory and Automatic Control at the University of Stuttgart. Her research focuses on quantitative dynamic modeling approaches and graph-theoretic analysis methods for cellular regulatory networks. She is member of the excellence cluster 'Simulation Technology' at the University of Stuttgart.

Ramesh Ram received the B.Tech degree in Information Technology from Anna University, Chennai, India, in 2005. He is currently working toward the Ph.D degree majoring in Information Technology, Bioinformatics and Computational Biology at Gippsland School of IT, Monash University, Churchill. His research interests include system biology, genetic network modelling and inference, microarray data analysis, Grid computing and pattern recognition.

Dr. **Andre Ribeiro** is a senior researcher at the Computational Systems Biology Research Group, at Technical University of Tampere, Finland. He graduated in Physics, at Faculty of Sciences of Lisbon University in 1999 and obtained the PhD in Physics at Institute Superior Tecnico, Technical University

of Lisbon, Portugal in 2004. From 1997-98, he was a researcher at Dept. of Optoelectronics, Institute of Technologies of Information INETI, Portugal. From 2004-07, he was a Post Doctoral fellow under the supervision of Stuart A. Kauffman, at the Institute for Biocomplexity and Informatics of the University of Calgary, Canada, focusing on models of Genetic Networks and inference algorithms from expression data. Dr. Ribeiro is a reviewer for several scientific journals and member of the Portuguese Physics Society and of the Canadian Society for Systems Biology. His research interests include gene networks dynamics, information propagation in systems, general principles governing cells' dynamics and emergent behaviours.

Peter Robinson is a physician by training. He completed his medical education at the University of Pennsylvania followed by an internship at Yale. Currently he holds a research scientist position at the Institute for Medical Genetics of the Charité Berlin. Additionally, he has obtained a BA in Mathematics and a Master of Science in Computer Science from Columbia University in New York City. A main focus in his research has been to use mathematical and bioinformatic models to understand biology and hereditary disease. In addition to computational biology, he also does "wetlab" molecular genetics research in hereditary disease as well as in the molecular mechanisms of fracture healing.

Lisa Schramm is a Ph.D. student at the Department of Control Engineering at the Darmstadt University of Technology, Germany since 2007. Her research project is a cooperation with the Honda Research Institute Europe GmbH. She studied Electrical Engineering and Information Technology and specialized in Control Engineering at the Darmstadt University of Technology. She graduated in 2007 and received the degree of Diplom-Ingenieur. Her main research interests are systems biology, development of neural networks and artificial life.

Bernhard Sendhoff studied physics at the Ruhr-Universität Bochum, Germany, and the University of Sussex, U.K. In 1998, he received the Doctorate degree in physics from the Ruhr-Universität Bochum, Germany. From 1998-1999 he was a research assistant at the Institute for Neuroinformatics. From 1999 to 2002 he worked for Honda R&D Europe GmbH, Offenbach/Main, Germany in several positions last as Deputy Division Manager. Currently, he is Chief Technology Officer of the Honda Research Institute Europe GmbH in addition to being Head of the Evolutionary and Learning Technology Group. Bernhard Sendhoff is Honorary Professor of the School of Computer Science of the University of Birmingham and Professor of the Technical University Darmstadt, Germany. His current research interests include topics from systems biology and computational intelligence such as evolutionary system design and structure optimization of adaptive systems. Bernhard Sendhoff is a senior member of the IEEE, a member of the ACM, the European Neural Network Society (ENNS) and the Deutsche Physikalische Gesellschaft (DPG). He is author and co-author of over 125 research papers in journals and refereed conferences.

He is a member of several advisory boards and scientific committees.

Ellen M. Sentovich received the B.S., M.S., and Ph.D. degrees in Electrical Engineering and Computer Science from the University of California Berkeley. She has worked for Intel, INRIA, the French National Computer Science Research Laboratory, and most recently as a research scientist at Cadence Berkeley Laboratories. Her research interests include logic synthesis, system-level specification and design, synchronous system design, and computational biology. She has served on many technical

reviewing committees, was technical chair and general chair of ICCAD, the International Conference on Computer-Aided Design, and general chair of DAC, the Design Automation Conference.

Dr. **Lina A. Shehadeh** earned a B.S. in Microbiology, a Masters in Teaching Sciences, and a Ph.D in Complex Systems and Brain Sciences, all from Florida Atlantic University. She was a postdoctoral fellow in Pharmacology and is now a Research Assistant Professor at the Department of Medicine in the University of Miami Miller School of Medicine. She uses bioinformatics and molecular biology tools to understand how microRNAs and transcription factors regulate the process of angiogenesis in the progression to heart failure.

Till Steiner studied Electrical Engineering at the Darmstadt University of Technology, Germany, and at the NanYang Technological University, Singapore. He graduated in 2006 from Darmstadt University of Technology and received the degree of Diplom-Ingenieur. Since 2006, he works as a scientist at the Honda Research Institute Europe GmbH. His research interests are based in the field of systems biology, with a focus on the simulation and analysis of gene regulation in multicellular development and multicellular methods for structure and topology optimization.

Marco Tomassini is a professor of Computer Science at the Information Systems Department of the University of Lausanne, Switzerland. He graduated in physical and chemical sciences in Mendoza, Argentina, and got a Doctor's degree in theoretical chemistry from the University of Perugia, Italy, working on computer simulations of condensed matter systems. His current research interests are centered around the application of biological ideas to artificial systems. He is active in evolutionary computation, especially spatially structured systems, genetic programming, and the structure of program search spaces. He is also interested in machine learning, evolutionary games, and the dynamical properties of networked complex systems. He has been Program Chairman of several international events and has published many scientific papers and several authored and edited books in these fields.

Rui-Sheng Wang is an Assistant Professor at School of Information, Renmin University of China. She received her Bachelor degree and Master degree in Applied Mathematics from Hebei University, respectively, in 1999 and 2002, Ph.D. degree in Operations Research and Control Theory from Academy of Mathematics and Systems Science, Chinese Academy of Sciences in 2005. Her current interests are within mathematical modeling and algorithm design in bioinformatics and systems biology, such as reconstruction of transcriptional networks, prediction and analysis of protein/ domain interactions, functional module detection in complex and biological networks.

Yong Wang is an Assistant Professor at Institute of Applied Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He received his Ph.D. degree in Operations Research and Control Theory from Academy of Mathematics and Systems Science, Chinese Academy of Sciences in 2005, Master degree in Operations Research and Control Theory from Dalian University of Technology in 2002, and Bachelor degree in Mathematics and Physics from Inner Mongolia University in 1999. His current interests include mathematical modeling and algorithm analysis in bioinformatics.

Frank Wimberly has worked in the area of scientific computing for over 35 years. He received a Ph.D. from the University of Pittsburgh, where he worked on problems in finite element methods.

He was a charter member of the faculty of the Robotics Institute at Carnegie Mellon University, after which he joined the Heinz School of Public Policy and Management at CMU as an Assistant Professor of Information Systems. There, he taught artificial intelligence and data communications while doing research in spatial diffusion modeling. He then spent several years as scientific applications coordinator at the Pittsburgh Supercomputing Center, where he published papers in the area of parallel computing. In his last position at CMU, Dr. Wimberly developed software for projects in statistical causal reasoning using Bayesian networks and structural equation models, and in symbolic logic. He has also worked in industrial positions at Bell Telephone Laboratories, Westinghouse Corporation, and BiosGroup.

Yu Xia is an Assistant Professor of the Bioinformatics Graduate Program and the Department of Chemistry in Boston University. He is also affiliated faculty of the Center for Advanced Genomic Technology in Boston University. Previously he was a Jane Coffin Childs Fellow in the Department of Molecular Biophysics and Biochemistry at Yale University. He received his Ph.D. in Chemistry from Stanford University, and B.S. in Chemistry (major) and Computer Science (minor) from Peking University. His current research interest is computer modeling of proteins encoded in the genome.

Dong Xu is James C. Dowell Professor and Chair of Computer Science Department, with appointments in the Christopher S. Bond Life Sciences Center and the Informatics Institute at the University of Missouri-Columbia. He obtained his Ph.D. from the University of Illinois, Urbana-Champaign in 1995 and did two-year postdoctoral work at the US National Cancer Institute. He was a Staff Scientist at Oak Ridge National Laboratory until 2003 before joining University of Missouri. His research includes protein structure prediction, high-throughput biological data analyses, computational proteomics, in silico studies of plant, microbes, and cancer.

Xiang-Sun Zhang is a Full Research Professor at Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He was graduated from Department of Applied Mathematics, Chinese University of Science and Technology in 1965. He is Honorary President of Operations Research Society of China. He has extensive experience in operations research, including optimization theory and application, artificial neural networks, management information system (MIS) theory and application. His current interests are in the application of operations research methods in bioinformatics.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y