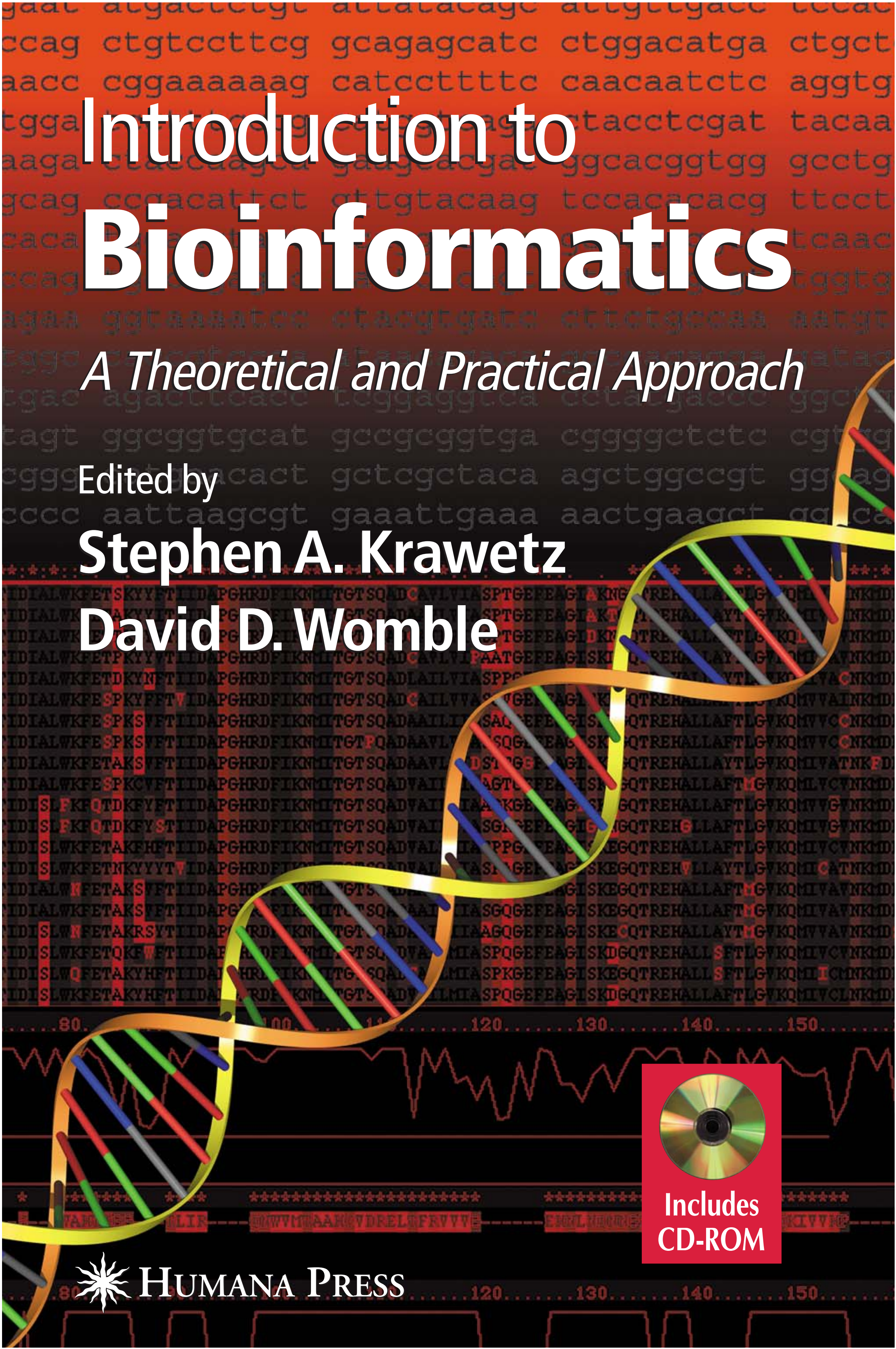


Introduction to **Bioinformatics**

A Theoretical and Practical Approach

Edited by

Stephen A. Krawetz

David D. Womble

Includes CD-ROM

Humana Press

Introduction to Bioinformatics

Introduction to Bioinformatics

A Theoretical and Practical Approach

Edited by

Stephen A. Krawetz, PhD

*Wayne State University School of Medicine,
Detroit, MI*

and

David D. Womble, PhD

*Wayne State University School of Medicine,
Detroit, MI*

Humana Press
Totowa, New Jersey

999 Riverview Drive, Suite 208
Totowa, New Jersey 07512

humanapress.com

All papers, comments, opinions, conclusions, or recommendations are those of the author(s), and do not necessarily reflect the views of the publisher.

This publication is printed on acid-free paper. ∞
ANSI Z39.48-1984 (American Standards Institute) Permanence of Paper for Printed Library Materials.

Production Editor: Mark J. Breaugh.

Cover design by Patricia F. Cleary and Paul A. Thiessen.

Cover illustration by Paul A. Thiessen, chemicalgraphics.com.

For additional copies, pricing for bulk purchases, and/or information about other Humana titles, contact Humana at the above address or at any of the following numbers: Tel.: 973-256-1699; Fax: 973-256-8341; E-mail: humana@humanapr.com, Website: humanapress.com

Printed in the United States of America. 10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging in Publication Data

Introduction to bioinformatics : a theoretical and practical approach / edited by Stephen
A. Krawetz and David D. Womble.
p. ; cm.
Includes bibliographical references and index.
ISBN 1-58829-064-6 (alk. paper) (HC); 1-58829-241-X (PB); 1-59259-335-6 (e-book)
1. Bioinformatics. I. Krawetz, Stephen A. II. Womble, David D.
[DNLM: 1. Computational Biology--methods. 2. Computer Systems. 3. Databases,
Genetic. 4. Genomics. 5. Sequence Analysis, DNA. 6. Software. QH 506 I646 2002]
QH 507 .I575 2002
570'.285--dc21 2002190207

Preface

As the sequencing phase of the human and other genome projects nears completion, we are faced with the task of understanding how the vast strings of Cs, As, Ts, and Gs encode a being. With the recent advent of microarrays and other high throughput biologic technologies, we have moved from trying to understand single molecules and pathways to that of integrative systems. We are only beginning to grasp the questions we can ask as we are now challenged to understand these large in silico, in vitro, and in vivo data sets. The new field of Bioinformatics was born of a series of meetings among "wet-bench" scientists, in the early 1980s, to meet this challenge. With the recruitment of mathematicians, computer scientists, statisticians, and astrophysicists to this field, we have now begun to design and implement some of the basic tools that will enable data integration and multidimensional analyses of these varied but unified data sets. For those new to Bioinformatics, this cross pollenization of the Life, Physical, and Theoretical sciences wants for a common language. With this in mind, *Introduction to Bioinformatics: A Theoretical and Practical Approach* was written as an introductory text for the undergraduate, graduate, or professional.

At once, this text provides the physical scientist, whether mathematician, computer scientist, statistician or astrophysicist, with a biological framework to understand the questions a life scientist would pose in the context of the computational issues and currently available tools. At the same time, it provides the life scientist with a source for the various computational tools now available, along with an introduction to their underlying mathematical foundations. As such, this book can be used as a bridge toward homologation of these fields. By bringing these disciplines together we may begin our journey toward understanding the nuances of the genetic code.

Introduction to Bioinformatics: A Theoretical and Practical Approach is divided into four main sections. The first two sections are well suited to the physical scientist who is new to studying biological systems. They provide the biological vocabulary, i.e., an overview of the various biological processes that govern an organism and impact health. The first section, *Biochemistry, Cell, and Molecular Biology*, describes basic cellular structure and the biological decoding of the genome. *In silico* detection of the promoter elements that modulate genome decoding is also explained. The second section, *Molecular Genetics*, will lead the reader through a discussion of the long range regulation of genomes, the *in silico* detection of the elements that impact long range control, and the molecular genetic basis of disease as a consequence of replication. Clinical human genetics and the various clinical databases are reviewed, followed by a discussion of the various issues within population genetics that can be used to address the question: "How do we evolve as we respond to our environment?"

The third section, *The UNIX Operating System*, was written for the life scientist, to demystify the UNIX operating system that is commonly used to support advanced computational tools. Along with understanding the installation and management of UNIX-based software tools, examples of command line sequence analyses are presented. These chapters should enable the life scientist to become as comfortable in a command line environment as in the Graphical-User Interface environment.

The *Computer Applications* section provides a common area for the physical and life scientist to meet. The management and analysis of DNA sequencing projects is presented, along with a review of how DNA can be modeled as a statistical series of patterns. The latter forms the basis of most protein and nucleic acid sequence analysis routines. These considerations are followed by a discussion of the various genome databases, the representation of genomes, and methods for their large scale analyses. This culminates in addressing the question: "Can I learn about my sequence from what is known about a similar sequence?" To directly answer this question a discussion of the various methods of pattern discovery follows, including basic multiple sequence alignment to identify both functionally and structurally related components. The accompanying protein visualization chapter outlines how these tools can aid in predicting structures that often represent homologous segments from evolutionarily conserved gene families. This final section concludes with a review of how multiple sequence alignment can be used to infer both functional and structural biological relationships. In closing, the final chapters of the book review the new field of Transcription Profiling, examining the current state of analysis software for systems biology. We conclude our journey with a discussion of the *in silico* analysis and prediction of patterns of gene expression that will ultimately guide our understanding of living systems.

Though the text provides a detailed description and examples, the CD supplement also contains a complete set of illustrations from each chapter, many of which are present in color. This provides a visual resource for both the student and the teacher that should prove invaluable for those of us preparing our next Bioinformatics lecture or seminar. In addition, several full version and limited trial versions of the programs that are discussed in the text are included. These encompass a broad spectrum, from DNA sequencing project management to microarray analysis, offering the reader the opportunity to easily access some of the software tools that are discussed. It is our hope that the current and next generation of physical and life scientists will use these resources as a springboard to help us move forward in the important quest for an integrated understanding of our physical being.

Stephen A. Krawetz
David D. Womble

Contents

B. Managing Bioinformatics Tools

C. Command Line Sequence Analysis

Part IV. Computer Applications:

A. Management and Analysis of DNA Sequencing Projects and Sequences

B. The Genome Database: ***Analysis and Similarity Searching***

C. Identifying Functional and Structural Sequence Elements

D. Analysis of Gene Expression: *Microarrays and Other Tools*

Part V. Appendices:

Contributors

Jill S. Barnholtz-Sloan • *Wayne State University School of Medicine and Barbara Ann Karmanos Cancer Institute, Detroit, MI*

Linda B. Bloom • *Department of Biochemistry and Molecular Biology, University of Florida, Gainesville, FL*

James K. Bonfield • *Laboratory of Molecular Biology, Medical Research Council, Cambridge, UK*

Broňa Brejová • *Department of Computer Science, University of Waterloo, Waterloo, ON, Canada*

Peter J. Bridge • *Department of Medical Genetics, Alberta Children's Hospital, Calgary, AB, Canada*

Bryon Campbell • *Van Andel Institute, Grand Rapids, MI*

Anthony V. Cox • *The Sanger Centre, Wellcome Trust Genome Campus, Cambridge, UK*

Paromita Deb-Rinker • *Department of Genetics, The Hospital for Sick Children, Toronto, ON, Canada*

Sorin Draghici • *Department of Computer Science, Wayne State University, Detroit, MI*

Brian Fristensky • *Department of Plant Science, University of Manitoba, Winnepeg, MB, Canada*

David J. Heard • *Department of Bioinformatics, ExonHit Therapeutics SA, Paris, France*

Teruyoshi Hishiki • *Biological Information Research Center, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan*

Jon Holy • *Department of Anatomy and Cell Biology, University of Minnesota School of Medicine, Duluth, MN*

Bradley C. Hyman • *Department of Biology, University of California at Riverside, Riverside, CA*

David P. Judge • *Department of Biochemistry, University of Cambridge, Cambridge, UK*

Jerzy Jurka • *Genetic Information Research Institute, Mountain View, CA*

Ben F. Koop • *Department of Biology, Center for the Environment, University of Victoria, Victoria, BC, Canada*

Jeffrey A. Kramer • *Monsanto Life Sciences, St. Louis, MO*

Stephen A. Krawetz • *Wayne State University School of Medicine, Detroit, MI*

Ming Li • *Department of Computer Science, University of California at Santa Barbara, Santa Barbara, CA*

Eric Martz • *Department of Microbiology, University of Massachusetts at Amherst, Amherst, MA*

JOHN R. MCCARREY • *Department of Biology, University of Texas at San Antonio, San Antonio, TX*

THOMAS W. O'BRIEN • *Department of Biochemistry and Molecular Biology, University of Florida, Gainesville, FL*

JOHN D. OFFERMAN • *Life Sciences Software Inc., Long Lake, MN*

KOUSAKU OKUBO • *Medical Institute of Bioregulation, Kyushu University, Fukuoka, Japan*

DANIEL A. RAPPOLEE • *Department of Obstetrics/Gynecology, Wayne State University, Detroit, MI*

C. A. RUPAR • *Biochemical Genetics Laboratory, Division of Clinical Biochemistry, CPRI, London, ON, Canada*

WOJCIECH RYCHLIK • *Life Sciences Software Inc., Long Lake, MN*

SITTICHOKE SAISANIT • *Department of Bioinformatics, Genetics, and Genomics, Hoffman-La Roche Inc., Nutley, NJ*

STEPHEN W. SCHERER • *Department of Genetics, The Hospital for Sick Children, Toronto, ON, Canada*

GAUTAM B. SINGH • *Department of Computer Science and Engineering, Oakland University, Rochester, MI*

RODGER STADEN • *Laboratory of Molecular Biology, Medical Research Council, Cambridge, UK*

JAMES W. STALKER • *The Sanger Centre, Wellcome Trust Genome Campus, Cambridge, UK*

STEVEN M. THOMPSON • *School of Computational Science and Information Technology, Florida State University, Tallahassee, FL*

TOMÁŠ VINAR • *Department of Computer Science, University of Waterloo, Waterloo, ON, Canada*

THOMAS WERNER • *GSF-National Research Center for Environment and Health, Neuherberg, Germany*

DAVID S. WISHART • *Faculty of Pharmacy and Pharmaceutical Sciences, University of Alberta, Edmonton, AB, Canada*

DAVID D. WOMBLE • *Center for Molecular Medicine and Genetics, Wayne State University, Detroit, MI*

CATHY H. WU • *Department of Biochemistry and Molecular Biology, Georgetown University Medical Center, Washington, DC*

THOMAS P. YANG • *Department of Biochemistry and Molecular Biology, University of Florida, Gainesville, FL*

Part I

Biochemistry, Cell, and Molecular Biology

A. The Cell

1. **Nucleic Acids and Proteins:** ***Modern Linguistics for the Genomics and Bioinformatics Era***
2. **Structure and Function of Cell Organelles**
3. **Cell Signaling**

1 Nucleic Acids and Proteins

Modern Linguistics for the Genomics and Bioinformatics Era

Bradley C. Hyman

Introduction

Genomics and Bioinformatics have the power to transform all facets of society. From anthropology to agriculture, medicine to manufacturing, virtually all disciplines undeniably will be changed by these promising fields. The goal of genomics is to mine the genomes of all relevant organisms to identify genes and their encoded products that govern the biological reactions that provide fuels, food, fiber, and other materials essential for our health. In addition to feeding the burgeoning world population, genomics/bioinformatics-based discovery will lead us to safer and more nutritious foods; self-resistant crops; disease-resistant animals; foods with prolonged shelf life; an understanding of why pathogens are virulent; and novel bio-based "smart molecules"such as alternative fuels, pharmaceuticals, and environmental sensors. Access to the genetic codes of microbes, plants, and animals will enable a clearer understanding of how life evolved on our planet.

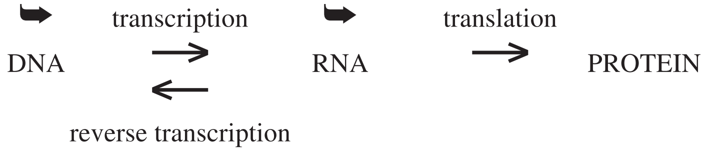
Training in genomics/bioinformatics requires a unique amalgam of skills in statistics, computer science (including algorithm development and database management), engineering, analytical chemistry, and of course, genetics and molecular biology. This chapter introduces venerable, fundamental concepts in molecular biology from a contemporary genomics/bioinformatics perspective using a language-based approach.

Building Definitions in the Genomic Era

What is Meant by Molecular Biology?

Many different kinds of molecules are components of living cells: carbohydrates, lipids, proteins, nucleic acids. All can be studied at the molecular level, including biosynthesis and assembly, atomic and molecular architecture, physiochemical properties, cellular targeting, and function. The common denominator among these avenues of investigation is the genes aligned along lengthy DNA strands. Stretched end-to-end, the genome would extend 2 to 3 meters in a normal human cell. Genes encode information for synthesizing molecules and their assembly into cellular structures. Molecular biology focuses on the structure and activity of these genes, which can be defined in two broad, fundamental ways:

1) The flow of information within a cell. One very famous descriptor of this activity is the *Central Dogma of Molecular Biology*. In its original concept, the *Dogma* was stated as seen in Scheme 1:



[Scheme 1]

By this scheme, DNA was considered to be self-replicating (➥), providing information in the form of a template for the precise duplication of one DNA molecule, the double helix, into two copies; the intracellular flow of information would continue to follow a strict path by *transcription* of a gene copy in the form of RNA, followed by interpreting the language of nucleic acids (nucleotides) into the language of proteins (amino acids) via *translation* of the RNA into a polypeptide chain. As seminal advances were made in the 1970s and 1980s, modifications were added to the central dogma that involved *reverse transcription* of RNA into DNA (typical of many dangerous retroviruses such as HIV-1, the AIDS virus), and self-replication of RNA molecules.

2) Heredity, or the flow of information between cells (as during cell division; binary fission in bacteria and mitosis in higher cells) and through the generations of an organism (typically mediated by eggs and sperm, produced through meiosis). The chromosomal basis of inheritance is founded on the process of DNA replication, and requires an intimate understanding of the functional architecture of the DNA molecule.

How Does Molecular Biology Help Us Define, in Contemporary Terms, Genomics and Bioinformatics?

The term *genome*, coined in the 1930s, refers to the complete set of genetic information found within an individual organism. Our understanding of molecular biology has enabled the establishment of a simple organizational hierarchy (*see* Fig. 1) that provides a useful scaffold for this discussion. We see that the genome is the all encompassing term for hereditary instructions, whereas the *nucleotide* is the fundamental chemical building block for the genetic material. Therefore, the genome is composed of the entire collection of nucleotides polymerized into long DNA strands. In the human genome, approx 3×10^9 nucleotides comprise one copy of our genome, condensed into each egg and sperm cell. This hierarchy can be annotated for humans by adding additional numerical values (*see* Fig. 2) to illustrate the interrelationship between the genome and other molecular units that collectively define our genetic material.

In Fig. 2, we find that each copy of the human genome is subdivided into 23 chromosomes. (When an egg and sperm unite during fertilization, the newly formed zygote contains 46 chromosomes, or two genome copies; most cells in our body are in this *diploid* state; eggs and sperm in the offspring remain *haploid* with one genomic copy). Individual chromosomes contain one DNA double helix, so each haploid genome is composed of 23 individual DNA duplexes. Distributed among these 23 helicies are an estimated 30,000 genes (although human gene approximations have ranged from 30,000–100,000). Nucleotides are bonded together to create the 23 indi-

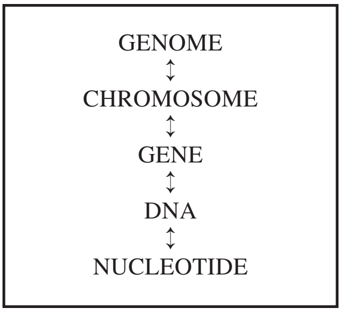


Fig. 1. Simple organizational hierarchy of genetic material.

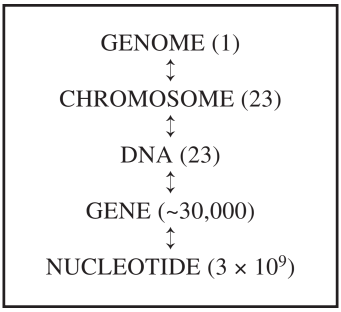


Fig. 2. Numerically annotated hierarchy of genetic material.

vidual DNA chains. Sizes of the 23 chromosomes range from 50 × 10^6 to greater than 250 × 10^6 nucleotides.

This hierarchy now allows us to define a "completely sequenced genome." Using high-throughput molecular technology and robotics in concert with sophisticated computer algorithms, it is now possible to assemble the precise order of the four nucleotide building blocks that comprise the single DNA helix within each chromosome. When such sequences are available for all chromosomes in an organism, the genome has been sequenced. Genome sizes range from about 2 million nucleotides for single bacterial chromosomes to 100,000 × 10^6 for some amphibians and plants. Each report of a completely sequenced genome represents an explosion of data that adds to the rapidly expanding international databases.

How Can All This Data be Stored, Accessed, Manipulated, Managed, and Analyzed?

Here, computer science and molecular biology partner into the new and rapidly emergent field of *Bioinformatics*. It is the goal of Bioinformatics to make sense of nucleotide sequence data (and for proteins, or *Proteomics*, amino acid sequence data). Assume that as we read this sentence, there are no spaces between words nor punctuation marks delimiting the boundaries of written thought. This same sentence might appear as:

assumethataswereadthissentencetherearenospacesbetweenwordsnorpunctuationmarks delimitingtheboundariesofwrittenthought

If this were a DNA sequence written in the language of the four nucleotide building blocks, or a protein sequence inscribed in the language of the 20 amino acids, informatic methodologies would be employed to extract and make sense of information encrypted in what superficially appears as a nonsense string. Bioinformatics as applied to DNA sequences would be exploited to find individual genes in the form of protein coding sequences (exons), expanses of nucleotides that might interrupt gene regions (introns), domains within the DNA that might control the expression of individual genes (e.g., promoters, enhancers, silencers, splice sites), repeated elements (insertion sequences and transposons in prokaryotes; micro- and mini-satellites in eukaryotic genomes), and other elements important for chromosome and gene maintenance. For proteins, identifying important domains within polypeptides, such as catalytic active sites, substrate binding sites, regions of protein-protein interaction, and the prediction of protein-folding pathways are important applications of bioinformatics. Exhuming this information is often conducted by aligning unknown nucleotide (or amino acid) strings with well-understood expanses of DNA or protein sequences to assist in the identification and determination of functional architecture. One popular avenue of research within Bioinformatics is the development and implementation of sophisticated alignment algorithms for the purpose of mining information from DNA and protein sequences.

The Language of DNA in the Genomics Era: Nucleotides and the Primary Structure of Nucleic Acid

This chapter is written in the English language that is composed of a 26-letter alphabet. Users of the language string letters into words, words into sentences, sentences into paragraphs, and so forth. The precise order of the letters conveys definition and meaning. Similarly, chains of DNA (and RNA) are polymers of four different chemical letters, or *nucleotide bases*; the precise order of polymerization is called the *primary structure* of nucleic acid and embedded within the primary structure is the definition of gene content.

The molecular structure of nucleotides also dictates important chemical properties of DNA. Nucleotides are composed of three chemically distinct precincts that confer functionality (*see* Fig. 3). These include a deoxyribose (DNA) or ribose (RNA) sugar, phosphate groups bonded to the 5' carbon in the deoxyribose (or ribose) sugar, and one of four nitrogenous bases (B; Fig. 3) that are attached to the 1' carbon. For DNA, the bases are: the purines, adenine (A) and guanine (G); the pyrimidines, cytosine (C) and thymine (T). Uracil (U), a pyrimidine that replaces T in RNA. When the complete DNA sequence of a genome is reported, it is actually the primary structure, or precise polymerization order, of the nitrogenous bases that are published as a simple string of As, Gs, Cs, and Ts. Sugars and phosphate groups remain invariant in the DNA chain (*see* Fig. 4).

See companion CD for color Fig. 4

The phosphate groups and deoxyribose (or ribose) sugars are highly polar and confer upon DNA (and RNA) the property of solubility in aqueous environments such as the interior of cells. At physiological pH, the phosphate group is ionized (deprotonated) conferring a net negative charge onto the nucleic acid polymer (*see* Fig. 4). In contrast, the nitrogenous bases are nonpolar entities that are “driven” into seclusion, away from aqueous environments. The interior of a DNA *double helix* provides such an environment; it is these hydrophobic forces, along with the additive Van der Waals interactions (0.1–0.2 kcal/mole) among the bases now stacked in the interior of the helix, that help stabilize the double helical structure. However, it is the unique arrangement of atoms within the nitrogenous bases that provides most of the stability to the double helix. When A is juxtaposed with T, and G is adjacent to C, the opportunity now exists for hydrogen bonding between hydrogen atoms of lower electronegativity resulting in a partially positive character, and oxygen atoms of high electronegativity with a partially negative character (*see* Fig. 5). This pairing behavior is often referred to as complementarity between bases (A and T, G and C). Hydrogen bonds (1–10 kcal/mole) formed precisely between the two *complementary* base pairs (A and T; G and C) stabilize the association between the two DNA chains resulting in the double helix.

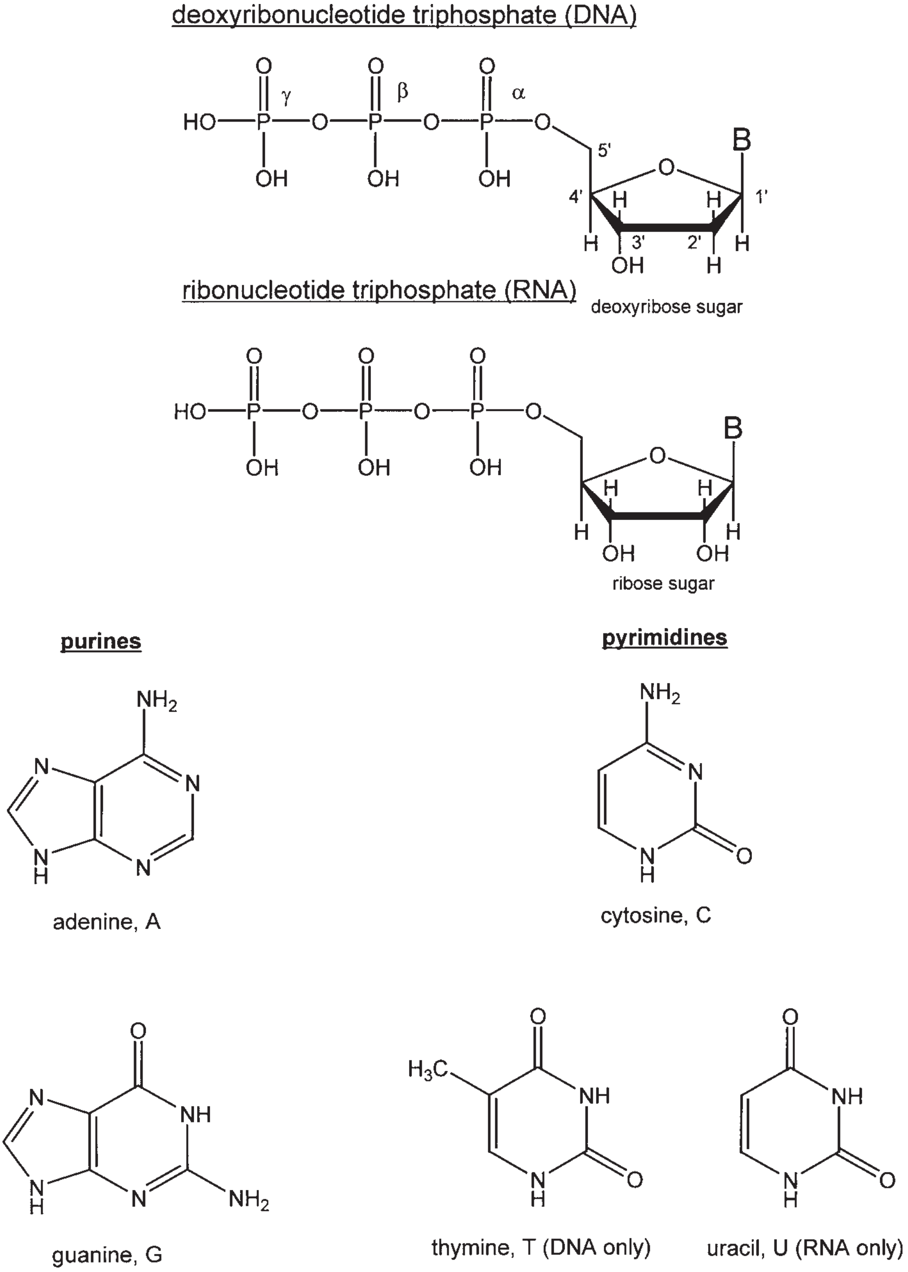


Fig. 3. Molecular structure of nucleotides, the building blocks of DNA and RNA. The 1' → 5' numbering convention used to designate carbons in the deoxyribose and ribose sugars is annotated only on the deoxyribose sugar. "B" extending from the 1' carbon of deoxyribo- and ribonucleotides represent the purine and pyrimidine nitrogenous bases. Designation of the different phosphate groups (α, β, γ) are depicted on the deoxyribose nucleotide.

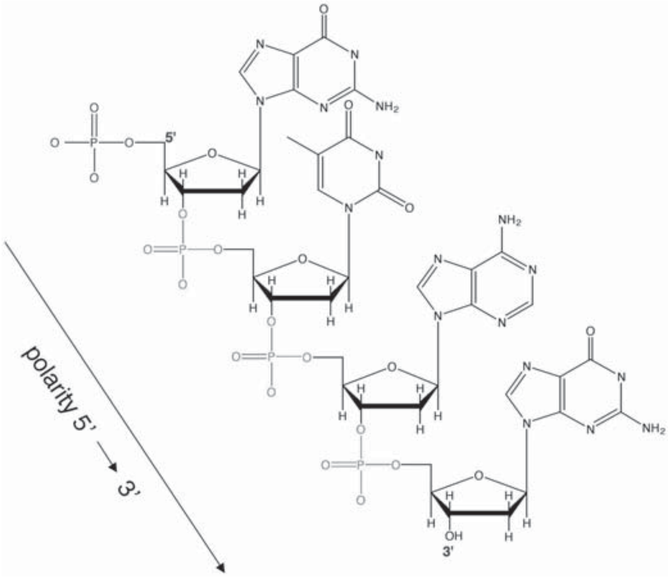


Fig. 4. Molecular structure of the tetranucleotide GTAG. The H present at 2' carbon of each sugar indicates that this is a DNA chain composed of deoxynucleotides. Phosphates involved in cementing adjacent nucleotides together with a phosphodiester bond are shown in red (on CD). A single free, 5' phosphate is pictured in blue. A free 3'-OH group at the opposite end of the chain is also depicted in blue.

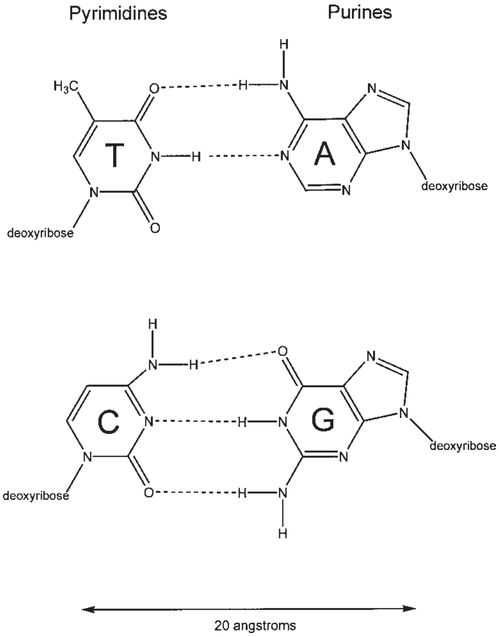


Fig. 5. Complementarity between specific nitrogenous bases. Pictured are the A-T and G-C base pairs present in a DNA double helix. (- - - -) represent hydrogen bonds between participating atoms. A purine juxtaposed with a pyrimidine after hydrogen bond formation generates a dimension of 20 angstroms, the width of a DNA double helix.

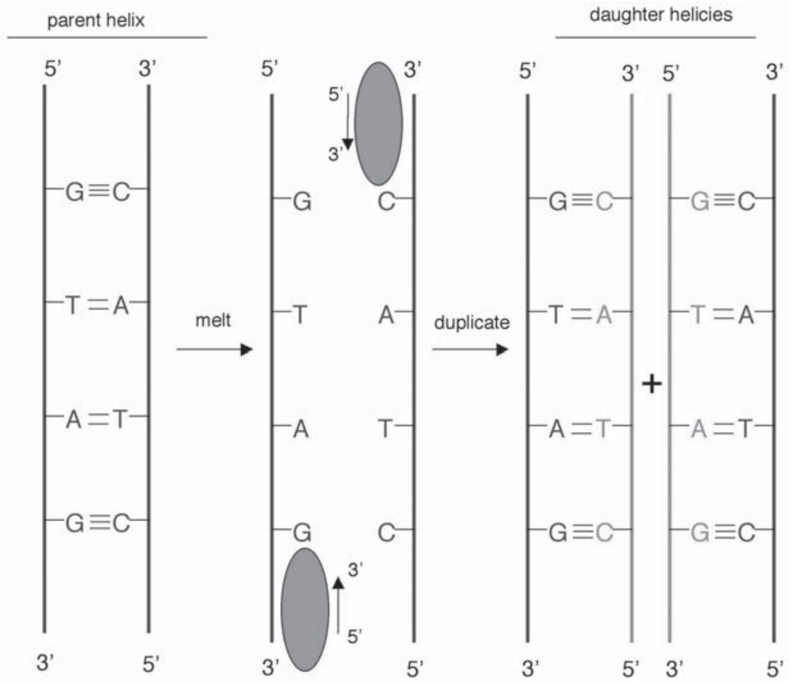


Fig. 6. Base pair complementarity is the basis for faithful duplication of the double helix. Chains of a DNA helix are pictured in an antiparallel configuration. Horizontal lines between the complementing bases denote hydrogen bonds (*see* Fig. 4). DNA polymerase (oval) polymerizes DNA in a 5' to 3' in an antiparallel direction on each strand. Each new daughter helix is composed of an original stand and a newly synthesized chain, indicative of semi-conservative replication.

As immediately recognized in their classic 1953 paper that first described the double-helical nature of the genetic material (*see* Suggested Reading), Watson and Crick explained

> "...the specific pairing we have postulated immediately suggests a possible copying mechanism for the genetic material."

See companion CD for color Fig. 6

What this statement was intended to convey is that each of the two strands in a DNA duplex, by virtue of their complementary, can act as an informational template to specify the primary structure of the second strand in a double helix (*see* Fig. 6). It is the principle of complementarity that provides the foundation for the faithful duplication of DNA, requisite to transmitting the genetic material at cell division and across generations. Semi-conservative DNA replication, in which the new daughter helices each contain one parental and one newly synthesized strand *(see* Fig. 6) is the initial step in the Central Dogma of Molecular Biology and at the mechanistic level provides the foundation for many procedures involved in genome sequencing This is discussed further in Chapter 4.

Nucleotides represent the basic chemical building blocks of the DNA or RNA chain. However, chromosomes are extremely long polymers of nucleotides cemented together. The glue between adjacent nucleotides in a nucleic acid chain is the *phosphodiester bond* (*see* Fig. 4). Formation of the phosphodiester bond in nature is catalyzed by the enzyme DNA polymerase (Figure 6, oval). Amazingly, the enzyme recognizes three substrates in a simultaneous fashion: 1) a free 3'-OH group of the nucleotide representing the growing end of a DNA chain (*see* Fig. 4, -OH); 2) the template DNA (the “opposite strand in a double helix) that provides instructions for the next nucleotide to be added in the form of complementary information (*see* Fig. 6, DNA strands); and 3) the appropriate nucleotide to be added to the growing end of the chain. The nucleotide to be added next into the polymer is in the form of a high energy *deoxynucleotide triphosphate* (dNTP; *see* Fig. 3). Note that the phospho-diester bond cementing two adjacent nucleotides together in the DNA chain contains but a single phosphate group (*see* Fig. 4, red). The energy released by the excision of two phosphate groups from the dNTP during polymerization is recruited by DNA polymerase to catalyze the formation of a a new covalent bond between the 3'-OH of the preceding nucleotide and the α-phosphate of the nucleotide that will be added (*see* Fig. 3). The overall chemical reaction for the polymerization of DNA can be written as shown in Equation 1:

$$(\mathrm{dNMP})_n + \mathrm{dNTP} \rightarrow (\mathrm{dNMP})_{n+1} + \mathrm{PP_i} \qquad \text{[Eq. 1]}$$

In Equation 1, n represents the number of nucleotides already polymerized in the DNA chain, dNMP represents any nucleotide polymerized into a DNA chain (note only one phosphate, a Monophosphate defines the precise structure of a phosphodiester bond), and PP$_i$ is inorganic phosphate, the two phosphates released from the deoxynucleotide triphosphate during addition of a nucleotide to a growing DNA chain.

DNA strands have a chemical directionality, or *polarity*. Specifically, the functional groups that make up the two ends of a DNA chain are different. We noted that a phosphate group resides at the 5' end nucleotide (*see* Fig. 4, phosphate group). That means that one end of the chain terminates in a phosphate group that is not otherwise occupied in a phosphodiester bond holding two adjacent nucleotides together. There will also be one 3'-OH group (*see* Fig. 4, -OH group) not engaged in cementing two nucleotides together in the polymer. This unoccupied -OH group will be found at the opposite end of the chain. Thus, the polarity of a DNA chain is arbitrarily defined as 5' → 3' and DNA sequences are recorded and read in this fashion. Hence, the sequence $^{5'}$P-GTAG-OH$^{3'}$ (shown in Fig. 4) would offer very different informational content than $^{5'}$P-GATG-OH$^{3'}$, the simple reversal of the above sequence. Importantly, the two extended chains in a double helix reveal an opposite, or *antiparallel* polarity (*see* Fig. 6). If we again take the tetranucleotide sequence: $^{5'}$P-GTAG-OH$^{3'}$ and now write this molecule in the form of a double helix, obeying all the rules of base pair complementarity, the correct depiction would be:

$$\begin{array}{c} {}^{5'}\mathrm{P\ –GTAG–O\ H}^{3'} \\ |\,|\,|\,| \\ {}^{3'}\mathrm{HO–CATC–P}^{5'} \end{array}$$

where the vertical lines denote hydrogen bonds between the complementing nucleotide bases in each strand.

Complementarity provides an important check for the precision of sequencing in the laboratory. When a nucleotide sequence is determined for one strand of a DNA duplex, the sequence of the opposite strand is easily predicted. Sequencing both strands of a gene region permits an infallible determination of a DNA sequence because of the cross-check provided by base pairing within the double helix. You may see the term "single pass" in some reports of genomic sequencing efforts, meaning that only one strand of the DNA was sequenced, but a single time. There is some inherent error as the sequence is not experimentally validated by its complement, but single-pass sequencing is useful for rapidly deducing the information content of a genomic region.

Transcription: Converting the Informatics of DNA into a Working RNA Copy

Transcription, or *gene expression* is the second step in the Central Dogma. Controlling gene expression at the level of transcriptional regulation is the subject of more papers published in Molecular Biology than any other topic. The process of transcription involves duplicating a gene sequence encoded in the DNA into an RNA copy. RNA, like DNA, is written in the language of nucleotides, although uracil (U) is substituted for thymine (T) in RNA molecules. The structural similarities between T and U (*see* Fig. 3) allow for either base to hydrogen bond with A. Chemical languages are not changed for RNA synthesis because both DNA and RNA are polymerized nucleotide chains. Information in DNA is simply copied, and this process is referred to as *transcription*. If the gene encodes a protein, the RNA draft is called *messenger*, or *mRNA*. The end-products of some genes are simply RNA copies, not protein. Typically, these are genes that encode *transfer RNAs (tRNAs)* and *ribosomal RNAs (rRNAs)*, both components of the translation apparatus.

Transcription, in biochemical terms, proceeds much like replication. The process can be described by the familiar Equation 2:

$$(\text{rNMP})_n + \text{rNTP} \rightarrow (\text{rNMP})_{n+1} + \text{PP}_i \qquad \text{[Eq. 2]}$$

The deoxyribonucleotides building blocks of DNA have been replaced with *ribonucleotides*, one of which carries the nitrogenous base uracil (U; *see* Fig. 3). The ribonucleotides contain a ribose sugar (*see* Fig. 3). When ribonucleotides are polymerized into RNA, the resulting chain is chemically unstable relative to its close DNA relative. Hydroxyl group (-OH) on the 2' carbon of the ribose sugar (*see* Fig. 3) in RNA can undergo a nucleophilic attack upon an adjacent phosphodiester bond in an RNA polymer and break the bond, fragmenting the RNA polymer. In contrast, deoxyribose sugars, the constituent of the DNA backbone, contain a chemically benign hydrogen at the 2' position (*see* Fig. 3). Although RNA is thought to be the Earth's first informational macromolecule, DNA has likely replaced RNA as the primary source of heritable genetic material because of its chemical stability.

RNA, like DNA, can assume various secondary structures. Whereas DNA is usually found as a rigid rod-like double helix as a consequence of hydrogen bonding between chains, RNA polymers typically exhibit intra-strand base pairing, G to C (as in DNA) and A to U (remember U replaces T in RNA). The result is a highly folded RNA chain. A classic example of RNA secondary structure is that assumed by tRNA, which folds into the well-recognized cloverleaf conformation (depicted in Fig. 7). RNA secondary structures provide important architectural features that influence function. Many RNA molecules, for example, are catalytic *ribozymes*, including the *peptidyltransferase* activity of the ribosome (which catalyzes the formation of peptide bonds). Its precise folding is required for this activity. Secondary structure at the 5' end of an mRNA molecule often influences the ability of the ribosome to engage the initiation of protein synthesis and provides a check point for controlling gene expression at the post-transcriptional level.

See
companion CD
for color Fig. 7

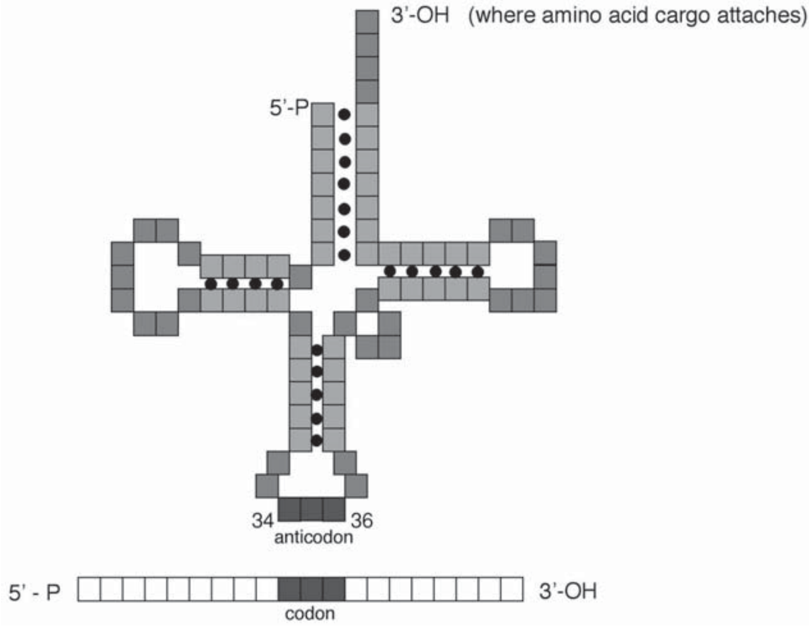


Fig. 7. A transfer RNA (tRNA) molecule acting as an interpreter between the language of nucleic acids (DNA and RNA) and the language of amino acids (proteins). A tRNA, pictured as a polymer of 76 individual ribonucleotides (individual squares), is folded into the universal cloverleaf structure by virtue of intra-molecular hydrogen bonding between complementing bases. The black dots denote base-pairing by hydrogen bonds to form *stems*. Gray squares are unpaired nucleotides, forming loops. The triplet anticodon nucleotides (34–36) are shown recognizing and interacting via hydrogen bonding with the appropriate triplet codon within an mRNA molecule.

Embedded within the primary sequence of DNA are sequences that control the initiation or termination of transcription. In effect, these are the primary regulators of gene expression. These signals are *promoters*, *enhancers*, *silencers*, *terminators*, and other sites along the DNA chains that are targets for DNA binding proteins. For example, promoters may be targets for RNA polymerases and other *general transcription factors* (TFs); enhancers provide docking sites for the proteins that are also needed to activate transcription, called *transcriptional activation factors* (TAFs); silencers bind proteins that inhibit or suppress transcription. One goal of genomics and bioinformatics initiatives is to identify these elements within genomic sequences, and address questions regarding commonalities among controlling elements for different genes. These strategies are helpful in providing a means to understand how genes residing at great distances along the DNA chain, or on different chromosomes, may be co-regulated in response to environmental cues.

The Language of Protein in the Genomics Era: Amino Acids and the Functional Architechure of Proteins

The central goal of genomics and bioinformatics is to understand the complete set of information encoded in a genome. Most of this information will reside in the ensemble of genes that dictates cellular functionality and the "assembly" of an organism. The end-product of most expressed genes is a protein, and the entire set of proteins elaborated by a specific cell type or by an organism is defined as the *proteome.*

The language of proteins is composed of an elaborate, 20 amino acid chemical alphabet. As with nucleic acids, the primary structure of a protein (the precise order of its amino acids) helps dictate the structure of a polypeptide. By analogy with nucleotides, we can dissect the chemical anatomy of amino acids to understand their role in directing protein structure. The typical structure of an amino acid is:

$$\begin{array}{c}\mathrm{R}\\ |\\ \mathrm{NH_2\text{-}CH\text{-}COOH}\end{array}$$

There are two important chemical features characteristic of this simple structure that are important for understanding the conformation of proteins. First, we note a chemical polarity to amino acids, with an *N*- or *amino- terminal* end (the NH_2 - amino group) and a *C*- or *carboxy-terminal* end (the -COOH or carboxylic acid group). Here we can draw an analogy with the chemical polarity of nucleotides defined by the 5' phosphate and 3'-OH groups. The NH_2-CH-COOH "backbone" is common among all 20 amino acids. Just as nucleotide bases are distinguished by one of four nitrogenous bases, amino acids differ from each other by the presence of one of 20 *side chains*, or *R-groups*. These R-groups are the information content of amino acids when polymerized into proteins, just as the nitrogenous bases are the informational component of nucleic acid chains.

The 20 side chains can be catalogued into three major groups depending on whether the R-group is nonpolar, polar, or charged at physiological pH (+ charge = basic; - charge = acidic). Figure 8 catalogues all 20 amino acids in these groups. When polymerized into proteins, amino acids will dictate whether a portion of the polypeptide is soluble in an aqueous environment (hydrophilic) usually if the protein region is rich in polar or charged amino acids, or is repelled by a water-like environment (hydrophobic) if the protein chain is locally rich in nonpolar R-groups. By sequestering nonpolar amino acids within the interior of a globular protein, and exposing polar and charged amino acids to the exterior, proteins in an aqueous environment are forced to assume a three dimensional, or "tertiary" conformation.

See companion CD for color Fig. 9

Proteins involved in membrane function provide an excellent example of how the precise distribution of amino acids within a polypeptide polymer govern structure/function relationships within a biological system (*see* Fig. 9). Membranes are composed of lipids, nonpolar hydrocarbons that present a hydrophobic environment. Proteins embedded in membranes serve many functions, including communicating with the environment and providing channels for transit of essential metabolites into and out of cells. A typical membrane protein (*see* Fig. 9) contains a series of trans-membrane domains composed almost exclusively of nonpolar, uncharged amino acids (small darker dots) that seek a hydrophobic environment within the interior of a membrane, as well as segments that interact with the aqueous interior and exterior of cells comprised by runs of polar and charged amino acids (small lighter dots).

Amino acids represent the building blocks of proteins. Adjacent amino acids are glued together via a peptide bond (*see* Scheme 2 on page 17). Two adjacent amino acids would exhibit the following structure shown in Scheme 2:

As with the synthesis of DNA, polymerization of amino acids into proteins involves a condensation/dehydration reaction that results in a peptide bond. R1 and R2

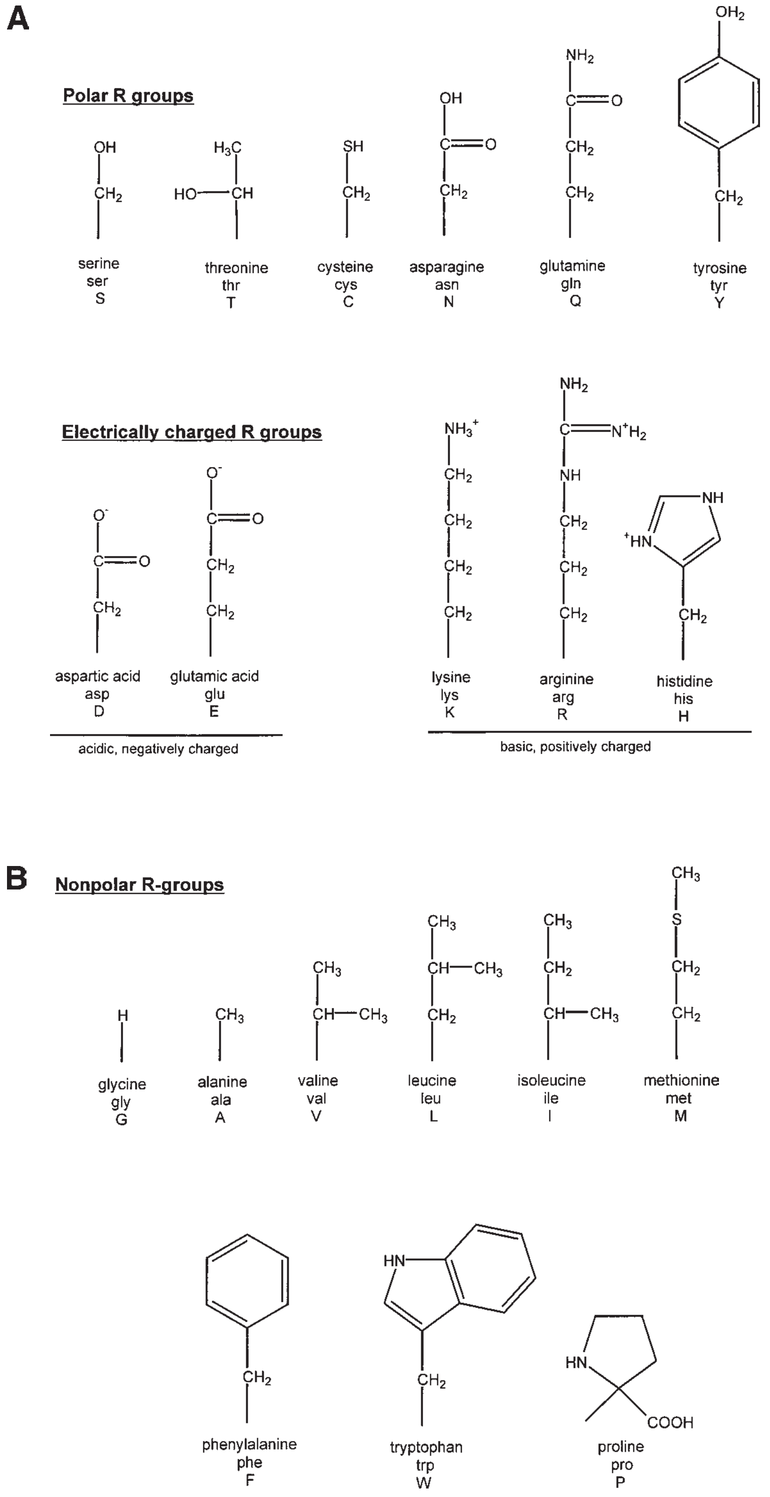

A
Polar R groups
OH
CH2
serine
ser
S
H3C
HO—CH
threonine
thr
T
SH
CH2
cysteine
cys
C
OH
C=O
CH2
asparagine
asn
N
NH2
C=O
CH2
CH2
glutamine
gln
Q
OH2
CH2
tyrosine
tyr
Y
Electrically charged R groups
O-
C=O
CH2
aspartic acid
asp
D
O-
C=O
CH2
CH2
glutamic acid
glu
E
acidic, negatively charged
NH3+
CH2
CH2
CH2
CH2
lysine
lys
K
NH2
C=N+H2
NH
CH2
CH2
CH2
arginine
arg
R
NH
+HN
CH2
histidine
his
H
basic, positively charged
B
Nonpolar R-groups
H
glycine
gly
G
CH3
alanine
ala
A
CH3
CH—CH3
valine
val
V
CH3
CH—CH3
CH2
leucine
leu
L
CH3
CH2
CH—CH3
isoleucine
ile
I
CH3
S
CH2
CH2
methionine
met
M
CH2
phenylalanine
phe
F
HN
CH2
tryptophan
trp
W
HN
COOH
proline
pro
P

Fig. 8. *(opposite page)* Molecular structure of the 20 amino acid side-chains (R-groups), listed according to chemical character (**A**, polar or **B**, charged). Beneath each structure is the name of the R-group, is three-letter designation, and its single letter designation. In an amino acid, the individual side-chain is attached to by a covalent bond to the common backbone NH_2 -CH-COOH, where C indicates the position of attachment.

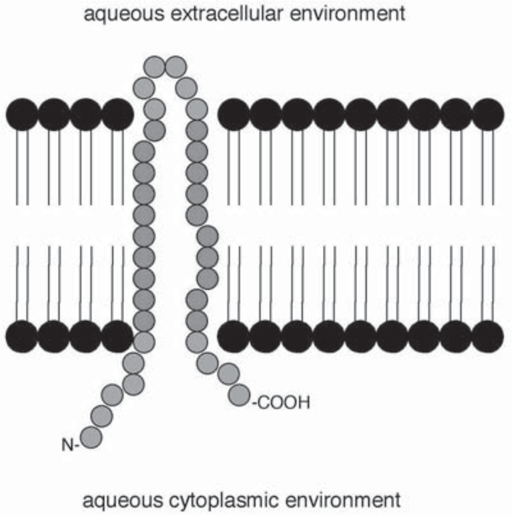
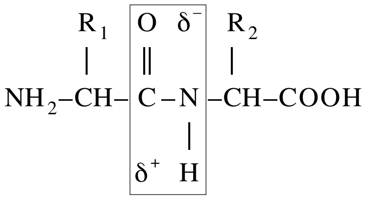

aqueous extracellular environment
-COOH
N-
aqueous cytoplasmic environment

Fig. 9. Membrane proteins: An example of how amino acid distribution dictates structure and function. Pictured in black are phospholipids that create the hydrophobic interior of a lipid bi-layer membrane. Small darker dots indicate nonpolar, hydrophobic amino acid constituents of a polypeptide chain. Small lighter dots represent polar, charged, amino acid residues that are capable of interacting with the aqueous, hydrophilic environments on either side of the cell membrane. Such trans-membrane protein domains typify proteins that serve as channels and as receptors to perceive external stimuli.

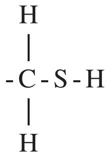


[Scheme 2]

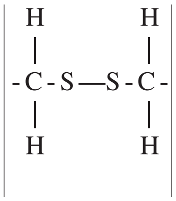
represent two different side chains. Each amino acid with their unique side-chain is typically given a three letter or a single letter designation (*see* Fig. 8). If the dipeptide depicted above is composed of the amino acids glycine and phenylalanine, the peptide would be written NH_2 -GLY-PHE-COOH or NH_2 - GP-COOH, or most often "GP." Note that proteins also have a chemical polarity, arbitrarily written from the N-terminal → C-terminal direction. Thus the dipeptide "GP" would be a very different from "PG." Chemical polarity serves an important role in the informational content of nucleic acids and proteins, as well as in how we annotate the primary structure of both macromolecules.

The oxygen and nitrogen atoms present in the peptide bond are electronegative. As such, the oxygen takes on a partial negative charge (δ^-) and the hydrogen covalently bonded to nitrogen a partial positive charge (δ^+). In a protein with hundreds of peptide bonds, this charge distribution is a recurrent theme, and hydrogen bonding (as in nucleic acids) between the partially charged H and O atoms occurs resulting in important secondary structures known as the α-*helix* and the β-*pleated sheet*. The α-helix is a coil, much like the a telephone handset cord; the β-pleated sheet adds a flattened, sheet-like topology to specific domains of the folded protein, much like a tightly routed "switchback" on a hiking trail.

One additional contributor to protein folding is offered by the chemical composition of the side-chain featured by the amino acid cysteine. The R-group for CYS (or C) is:



When two CYS residues in different regions of a folded protein chain find themselves in the vicinity of each other as a consequence of protein folding, oxidation allows a convalent bond (below) to form between the S atoms in each participating side chain, resulting in a disulfide bridge:



We observe that folded protein chains can become stabilized by the formation of disulfide bridges, contributing to the overall topography of the polypeptide. Unlike DNA, every individual protein assumes a unique three-dimensional, tertiary conformation that is necessary for catalytic activity (in the case for enzymes) or to play structural roles within the cell. Tertiary structure, then, is determined by a combination of factors that include the precise order of amino acids (the primary structure), formation of secondary structures in the form of α- helicies or β-pleated sheets and folding due to the generation of disulfide bridges, then driven by hydrophobic and hydrophilic interactions between amino acids and their environment. An even higher level of structure is achieved because many active proteins are not monomers (one properly-folded polypeptide chain), but subunits that must interact with other properly folded proteins to form an active *quarternary structure*. One popular example is the protein hemoglobin, the oxygen carrier protein in our red blood cells. Active hemoglobin is a tetramer comprised of two α and two β subunits in erythrocytes of adult humans. In isolation, these individual protein subunits are inactive, but when they come together they bind and carry oxygen.

Translation: Converting the Language of Nucleic Acids into the Language of Proteins

Proteins are composed of amino acids; however, the information for directing the precise order of amino acids within a peptide (the primary structure) is encoded within DNA and its constituent nucleotides. Conversion between any two languages requires a translator, and no less is true in the cell. The molecular interpreter is the *ribosome*, the cellular site where polymerization of amino acids into protein occurs. *Translation* is the final step in the Central Dogma.

The primary sequence of the nucleotides within DNA, and its RNA copies, can be considered a code that requires deciphering. Proteins are composed of 20 different amino acids, and the first question regarding the code is "How many nucleotides are required to specify one amino acid in a protein sequence?" This question led to the concept of the *coding ratio*: number of nucleotides required to encode one amino acid, or the number of nucleotides/amino acid.

There are four nucleotides (A,G, C, T). Thus if one nucleotide encoded one amino acid (coding ratio = 1), the genetic code could only accommodate 4 amino acids. This is not suffient to encode 20 amino acids. If the coding ratio were two, there are $4^2 = 16$ possible dinucleotide combinations, and 16 amino acids could be encoded. The requirement to encode 20 amino acid is not quite achieved with a doublet code. A coding potential of 64 amino acids (4^3) is achieved with a coding ratio of three, and hence the *triplet codon* (three contiguous nucleotides along the DNA or RNA chain) encodes the information for a single amino acid (*see* Table 1). The 64 codon carrying capacity of this code far exceeds the necessary requirement to encode 20 amino acids. However, the code is redundant; i.e., the same amino acid is often encoded by several different nucleotide codons that are synonyms. For example, the amino acid leucine (LEU) is encoded by six different triplets (*see* Table 1).

Three codons do not encode any amino acid. They are *termination* or *stop codons* that delimit the C-terminal end of the protein, where translation stops. Continuing with our analogy of languages, termination codons serve as the period to end a sentence.

The mRNA copy of the information encoded in DNA is translated into a protein in groups of three nucleotides. Translation typically commences at an AUG (methionine, MET) codon that also signals the initiation of translation. This sets the *reading frame*, one of three possible ways any mRNA molecule can be read in groups of triplets. To illustrate the concept of a reading frame, let us create an mRNA molecule as a sentence in the English language that consists of words with only three letters: "The fat cat ate the big rat." When read as triplets, this rendition makes sense to us. But read in triplets from a different start point, the sentence could also be read "..T hef atc ata tet heb igr at.." or even in a third frame as "..Th efa tca tat eth ebi gra t.." A triplet code implies there are three possible reading frames within a mRNA molecule. The AUG *start codon* establishes the correct reading frame, the one that makes sense. This is defined as an *open reading frame* that typically begins with a start codon (AUG) and ends with one of three termination codons (UAG, UGA, UAA). Gene discovery within genomic sequences, annotating genes, and using sequence information for alignments relies on these features of the nearly universal genetic code (*see* Table 1).

Clearly, a molecular adaptor is required to decipher each triplet codon and deliver the correct amino acid into the growing protein chain. This adaptor is a specific tRNA molecule of about 76–80 nucleotides in length (*see* Fig. 7). Typically residing at nucleotides 34–36 is a set of three continguous nucleotides, a triplet anticodon with base

Table 1
Universal Genetic Code

First	*Second*				*Third*
5′	*U*	*C*	*A*	*G*	*3′*
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	Tyr	Stop	A
	Leu	Ser	Tyr	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

pair complementarity to a codon along the mRNA chain (*see* Fig. 7). At the 3' end of this molecule the corresponding amino acid cargo is carried (*see* Fig. 7, Table 1). The mRNA and tRNAs congregate at the ribosome, a complex cellular organelle with multiple functions that include: 1) transiting along the mRNA chain in three nucleotide (codon) intervals; 2) capturing the tRNA dictated by the appropriate anticodon/tRNA combination; 3) catalyzing the hydrolytic removal of the amino acid from the tRNA adaptor and 4) condensing the same amino acid to the growing peptide by formation of a peptide bond. This same series of reactions occurs in a sequential fashion, three nucleotides at a time, for each codon found in the mRNA. Translation is an energetically expensive and complex process; three high energy molecules at ATP or GTP are invested to add each amino acid to a growing protein chain as further discussed in Chapter 5.

A Perspective

Modern genomics and bioinformatics is really a new kind of linguistics. Nature, through dynamic evolutionary forces, inscribes genetic information in the form of long nucleotide chains, or genomes. Deciphered by the cell using steps in the Central Dogma, the end-products of sensible nucleotide strings (the genes) are highly versatile proteins whose three-dimensional structure dictates function. The universal genetic code was *cracked* over four decades ago. Yet, how genomes encode traits that make each organism and species different from each other and each individual within a species unique, remained encrypted information until now. With the advent of DNA and protein methodologies, genomes no longer present themselves as molecular hieroglyphics.

Glossary and Abbreviations

Amino Acids Building blocks of protein chains.

Amino Terminal End The end of a protein chain with a free amino group; sometimes called the N-terminal end.

Antiparallel DNA chains in a double helix polymerized in the opposite polarity.

β-Pleated Sheet One of two major secondary strcuture conformations assumed by protein chains.

Bioinformatics Analysis of nucleic acid or protein sequence data.

Coding Ratio Number of nucleotides required to encode a single amino acid .

Complementary Describes the precise base pairing of A with T and G with C by hydrogen bonding.

Cracked Decoded.

Diploid A cell containing two complete sets of chromosomes or genetic information, as found in most body cells.

dNTP Deoxynucleotide triphosphate, the building block of DNA chains.

dNMP deoxynucleotide monophosphate, the form of the building block that eventually becomes incorporated into DNA chains.

DNA Deoxyribonucleic acid.

DNA Replication The act of duplicating DNA from one double helix to two new helicies.

Double Helix The famous two stranded secondary structure of DNA.

Enhancer Binding site in DNA, upstream or downstream from a promoter, for transcription factors that enhance gene expression.

Gene Expression Converting a segment of DNA (gene) into a RNA copy.

General Transcription Proteins necessary to turn on transcription of gene factors.

Genome The complete set of genetic information found in an organism.

Haploid A cell containing one complete set of chromosomes or genetic information, as found in sperm and eggs.

α-Helix One of two major secondary structure conformations assumed by protein chains.

Messenger RNA RNA copy of the gene that encodes a protein.

MRNA Messenger RNA.

N-Terminal End The end of a protein chain with a free amino group; also called the amino terminal end.

Nucleotide The monomer building block of nucleic acid chains.

Nucleotide Bases One of five ring structures, A, G, C, T, U that provides information content to nucleic acid chains.

Open Reading Frame A nucleotide sequence beginning at AUG (the start codon), then read for many consecutive triplets, and ending with one of three stop codons.

Petidyltransferase Catalytic activity of ribosomes that create peptide bonds when amino acids are polymerized into proteins.

Phosphodiester Bond The bond that cements two adjacent nucleotides together in a DNA or RNA chain.

Polarity The chemical directionality of a nucleic acid or protein chain.

Primary Structure The precise order of nucleotide bases or amino acids in a nucleic acid or protein chain, respectively.

Promoter Binding site in the DNA for transcription factors and RNA polymerase; The on/off switch for gene expression.

Proteome Entire collection of proteins in a cell or organism.

Proteomics The study of the entire ensemble of proteins in a cell or organism.

Quarternary Structure Level of structural organization when two or more individual protein chains (subunits) interact to form an active complex.

R-Group One of 20 amino acid side chains that specify chemical properties to each amino acid.

Reading Frame Lengthy contiguous stretches within a gene that specify consecutive triplet codons without encountering a stop codon; this specifies the coding region of a gene; also known as an open reading frame.

Reverse Transcription Synthesizing DNA from an RNA template, as is the case for the AIDS virus.

Ribosomal RNA RNA that forms the structural and catalytic cores of ribosomes.

Ribosome Multisubunit organelle in all cells that is the site of protein synthesis.

Ribozymes RNA molecules that exhibit a catalytic activity.

RNA Ribonucleic acid.

rNTP Ribonucleotide triphosphate, the building block of DNA chains.

rNMP Ribonucleotide monophosphate, the form of the building block that eventually becomes incorporated into DNA chains.

rRNA Ribosomal RNA.

Side Chain One of 20 different chemical groups that confer properties to different amino acids.

Silencer Binding site in DNA for proteins that suppress transcription (gene expression).

Start Codon The triplet AUG that specifies where protein synthesis begins and sets the reading frame.

Stop Codon One of three triplet codons, UAG, UAA, UGA that specify where an mRNA is to stop being translated into protein.

TAF Transcription activation factor.

Terminator Segment of DNA where transcription ends.

Termination Codon One of three triplet codons, UAG, UAA, UGA that specify where an mRNA is to stop being translated into protein.

TF Transcription factor.

Transcription Synthesizing RNA from a DNA template using RNA polymerase transcription activation factor accessory proteins needed for gene expression.

Translation Synthesizing a protein (polypeptide) from mRNA instructions with the ribosome.

Transfer RNA Small RNA molecule that adapts the language of nucleic acids to the language of proteins.

Triplet Codon A string of 3 consecutive nucleotides in an mRNA molecule that specifies one of the 20 amino acids.

TRNA Transfer RNA.

Suggested Readings

Basic Texts

Brown, T. A. (1999) Genomes, John Wiley and Sons, New York, NY, p. 472.
Lewin, B. (2000) Genes VII, Oxford University Press, Oxford, UK, p. 990.
Mount, D. W. (2001) Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, p. 564.
Russell, P. J. (2001) iGenetics, Benjamin Cummings, San Francisco, CA, p. 828.
Weaver, R. F. (1999) Molecular Biology, WCB/McGraw-Hill, Boston, MA, p. 788.

Some Primary Literature

(2001) The Human Genome Issue, Nature 409, 745–764.
(2001) The Human Genome Issue, Science 291, 1145–1434.
(2000) Breakthrough of the Year: Sequenced Genomes, Science 290, 2201–2372.
Watson, J. D. and Crick, F. H. C. (1953) Nature 171, 737–738.

2 Structure and Function of Cell Organelles

Jon Holy

Introduction

The myriad biochemical reactions that comprise life processes are too numerous and complex to be carried out entirely by simple diffusion-mediated interactions between enzymes and substrates. Instead, sequences of biochemical reactions must be efficiently organized and integrated with other sets of reactions by the cell. Two fundamental structural elements are used by eukaryotic cells to organize and integrate these reactions: membranes and a cytoskeletal system. An elaborate system of cellular membranes, in the form of the plasma membrane, membrane-bound organelles, and the nuclear envelope, has evolved to provide reaction surfaces and to organize and compartmentalize molecules involved in specific metabolic pathways. Other cytosolic biochemical reactions, as well as the organization of membranous organelles within the cell, are regulated by interactions with the cytoskeletal system. Consequently, enzymes and proteins involved in biochemical reactions can be located in the cytosol, within membranes, on the surfaces of membranes, within the interior of membrane-bound compartments, or in association with the cytoskeleton. The elaboration of these structural elements has allowed for the sophisticated level of biochemical integration that exists in living eukaryotic cells.

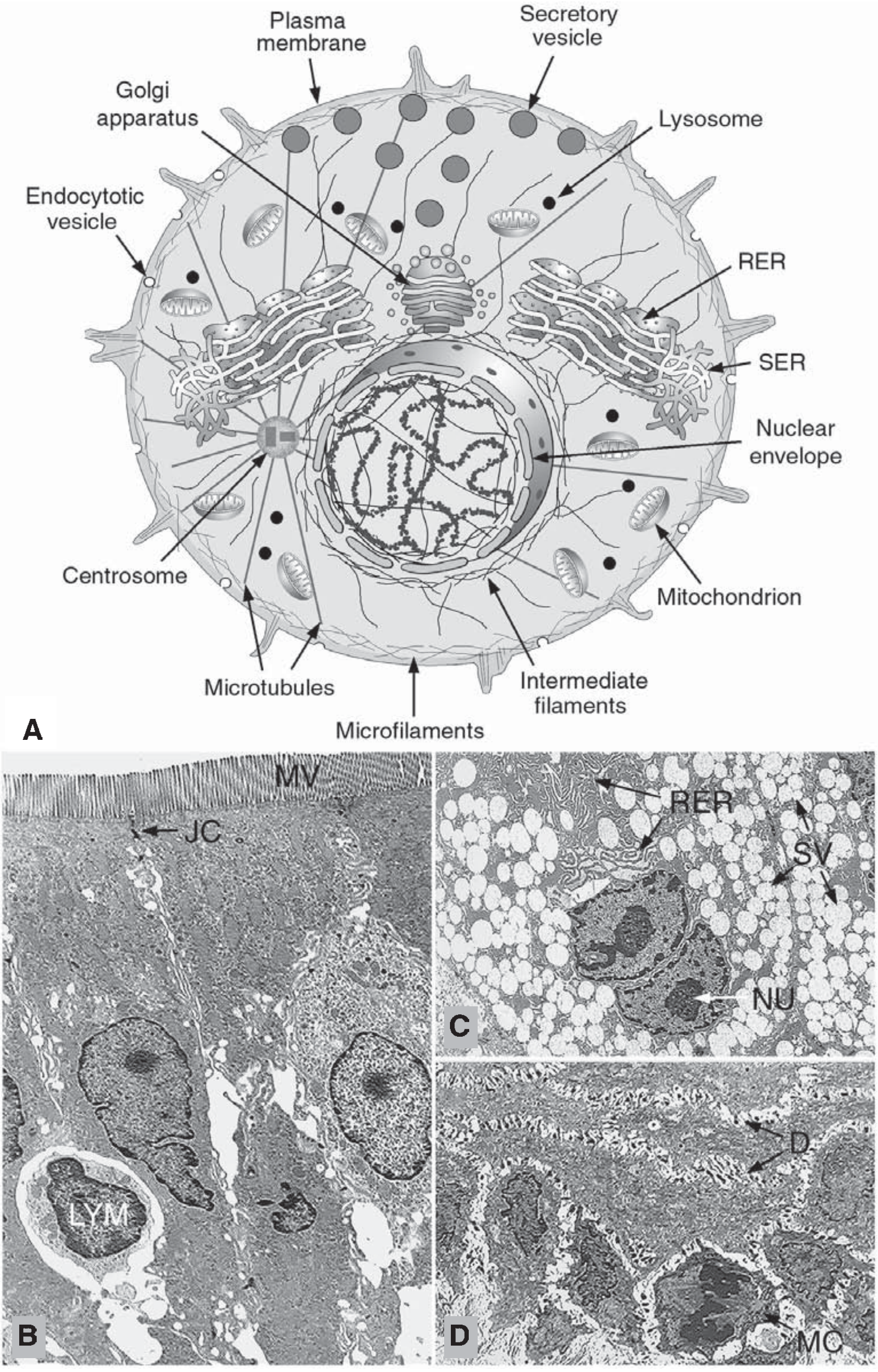
Over two hundred different types of cells are found in higher animals, including humans, and the interaction of these diverse cell types is responsible for the formation and functioning of tissues and organs. Different types of cells carry out specialized functions, but all cells face similar sets of challenges to exist. In general, cells must maintain a barrier against, and sensing mechanisms to interact with, their external environment; synthesize and recycle their structural and enzymatic components; repair physical or chemical damage; grow and reproduce; and generate energy for all of these activities. These generalized functions, as well as the more specialized functions of individual cell types, are all performed by cell organelles. Cell organelles perform the basic functions that allow cells to survive and replicate, and are dynamic entities that become modified to help specialized cells carry out specific functions. For example, all cells contain a cytoskeletal filamentous system that functions in maintenance of cell shape and allows for some degree of movement, but muscle cells contain far greater numbers of these filaments to carry out the contractile activity that comprises muscle activity.

Classically, the phrase *cell organelle* has been used to denote distinct membrane-bound structures that are readily visible by light or electron microscopy and possess characteristic morphological features that make them readily identifiable in essentially all eukaryotic cells. Such structures include the plasma membrane, ER, Golgi apparatus, lysosomes, peroxisomes, and mitochondria (Fig. 1). The structure and function of these organelles, as well as the cytoskeleton and nucleus, are described in this chapter. Because membrane structure plays fundamental roles in organelle function, the basic features of membrane organization will be considered first.

Membrane Structure

Cell membranes are composed of lipid and protein, which are assembled into two opposing layers called the lipid bilayer. Four major types of phospholipids and cholesterol comprise most of the lipid portion of the bilayer. The phospholipids include the choline-containing lipids phosphatidylcholine and sphingomyelin, and the amine-containing lipids phosphatidylethanolamine and phosphatidylserine. All of these phospholipids possess hydrophilic polar heads, and two hydrophobic fatty acid tails. The membrane bilayer represents an energetically favorable conformation of these lipids in that the tails associate with each other to form a hydrophobic environment in the center of the bilayer, with the polar heads facing outward to interact with the charged aqueous environment of the cytoplasm, organelle lumen, or extracellular space. The hydrophobic region resulting from the association of lipid tails creates a barrier to the passage of charged molecules, and only small uncharged molecules, or lipid-soluble molecules, can freely penetrate the lipid bilayer. Cholesterol, which is shorter and stiffer than phospholipids, can comprise up to about 50% of the total membrane lipid. The hydroxyl end of cholesterol interacts with the polar heads of phospholipid molecules, with the rest of the molecule in the same plane as the fatty acid tails of phospholipids. Its presence in membranes is thought to help prevent phase transitions by stiffening membranes at higher temperatures, while also maintaining membrane fluidity at lower temperatures. Although cholesterol is prevalent and equally represented

Fig. 1. *(opposite page)* Overview of cell organization. **(A)** Diagram of the major cell organelles, including the cytoskeleton and nucleus. This drawing depicts a single idealized cell, and so does not include the cell-cell and cell-ECM junctions elaborated by cells in tissues. **(B–D)** Examples of low-power electron micrographs of thin sections of rat tissues showing how cell organization reflects cell function. **(B)** Shows intestinal epithelial cells, which are modified to aid in the digestion and absorption of food. The apical membranes of these cells develop highly organized microvilli (MV), which are supported by bundles of microfilaments, to increase the surface area of these cells. The epithelial cells are bound to each other by junctional complexes (JC) consisting of a cluster of tight junctions, zonulae adherens junctions, and desmosomes. The tight junctions prevent material in the intestinal lumen from diffusing between cells into the body cavity, and the adherens and desmosomal junctions firmly anchor cells to each other. A migrating immune cell (LYM, lymphocyte) is also present in this section. **(C)** Shows salivary gland cells, which are specialized to produce and release large amounts of secretory glycoproteins. These cells contain extensive arrays of rough endoplasmic reticulum (RER), and become filled with secretory vesicles (SV). Two nuclei are present in this section, and both display prominent nucleoli (NU). **(D)** Shows cells of the esophageal epithelium, which are specialized to accomodate and resist mechanical stresses. These cells are constantly renewed by the mitotic activity of a basal layer of cells (MC, mitotic cell). As cells are produced and differentiate, they move toward the lumen of the esophagus, become flattened, and form extensive desmosomal connections (D) with neighboring cells.


Plasma membrane
Secretory vesicle
Golgi apparatus
Lysosome
Endocytotic vesicle
RER
SER
Nuclear envelope
Centrosome
Mitochondrion
Microtubules
Intermediate filaments
Microfilaments
A
MV
JC
LYM
B
RER
SV
NU
C
D
MC
D

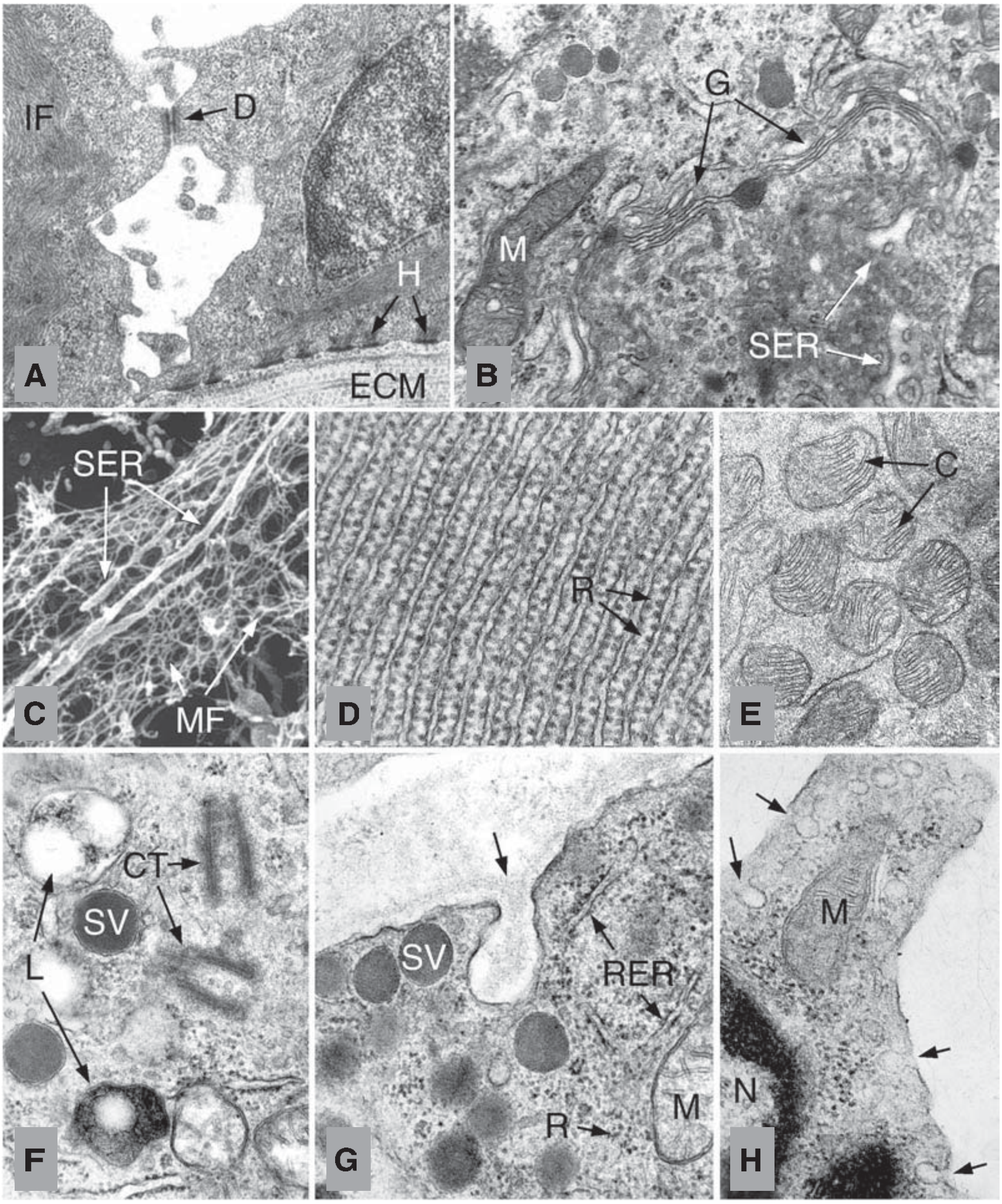


Fig. 2. Electron micrographs illustrating the structural features of various cell organelles. **(A)** Epithelial cells from a tadpole (*Rana pipiens*) tail; **(B)**, **(F)**, **(G)**, and **(H)** show endocrine cells from a rat pituitary gland; **(C)** sea urchin coelomocyte; **(D)** and **(E)** secretory cells from digenetic trematodes (*Halipegus eccentricus* and *Quinqueserialis quinqueserialis*). **(A)** Low-power electron micrograph of an epidermal cell, showing a number of cell-cell and cell-extracellular matrix junctions. These cells elaborate numerous desmosomes (D) and hemidesmosomes (H), and the cytoplasm is filled with prominent bundles of intermediate filaments (IF), which interconnect these junctions. ECM, extracellular matrix. **(B)** Cytoplasm of an endocrine cell, showing a mitochondrion (M), smooth endoplasmic reticulum (SER), and flattened cisternae of the Golgi apparatus (G). Also present are small clusters of

in both bilayers of a membrane, the major phospholipids are asymmetrically distributed, with higher concentrations of choline-containing phospholipids present in the noncytoplasmic layer (i.e., the layer facing organelle lumens and the layer of the plasma membrane facing the extracellular matrix), and higher concentrations of amine-containing phospholipids in the layer facing the cytoplasm.

In addition to lipid, membranes are also composed of protein. Membrane proteins are either classified as integral membrane proteins if they penetrate or are anchored in the bilayer, or as peripheral membrane proteins if they are just associated with the surfaces of the bilayer. Integral membrane proteins are difficult to remove from membranes, usually requiring disruption of the lipid bilayer (e.g., with detergents) to be released. Peripheral proteins are easier to remove from membranes, as they are generally held in place by protein-protein interactions. Integral membrane proteins can penetrate the bilayer completely a single time (single-pass proteins) or multiple times (multi-pass proteins). They can also be anchored in the membrane through covalent attachments to lipid molecules in the bilayer.

A number of membrane lipids and proteins are glycosylated. Glycosylation of membrane components takes place in the ER and Golgi apparatus. Because glycosylation occurs exclusively within the interior (or lumen) of these organelles, the sugar groups of glycoproteins and glycolipids all face toward the lumenal surface of membranes of organelles, and the extracellular matrix (ECM) side of the plasma membrane. Glycosylation of membrane lipids and proteins is thought to help protect membranes, and in the case of the plasma membrane, to help identify the cell and to assist in the adhesion of cells to the ECM.

Membrane lipids and proteins carry out a number of functions. In addition to serving as the structural framework of the membrane, they mediate the functions of all membranes of the cell. Membrane lipids can form specialized subdomains composed of specific lipid populations (lipid rafts) that appear to facilitate localized membrane function, and some membrane lipids are intimately involved in signal transduction events. Membrane proteins carry out a wide variety of functions, including serving as membrane channels, carriers, and pumps; transducing cytoplasmic and extracellular signals; targeting membranes to specific locations; and adhering membranes to each other and to the ECM.

Fig. 2—*(continued)*

free ribosomes, secretory vesicles, and part of the nucleus (upper left corner). **(C)** Low-voltage scanning electron micrograph of an extracted cell (the plasma membrane and soluble cytoplasmic proteins were removed by detergent), showing a few tubular extensions of SER embedded in a network of microfilaments (MF). This type of microscopy shows the surface features of organelles. **(D)** A dense array of RER from a secretory cell; note the high level of organization in the parallel alignment of cisternae. Attached ribosomes (R) appear as small granular bodies. **(E)** Section of cytoplasm containing a number of mitochondria. Their striped appearance is due to the invagination of the inner mitochondrial membrane, which forms cristae (C). **(F)** Endocrine cell cytoplasm, showing two small lysosomes (L), secretory vesicles (SV), and a centriole pair of a centrosome. In this section, pericentriolar material and attached microtubules are not clearly displayed. **(G)** Exocytosis in an endocrine cell. This micrograph shows the periphery of the cell, and the deep invagination of the plasma membrane indicates the site where a secretory vesicle has undergone exocytosis (arrow). A mitochondrion (M), some strands of RER, and clusters of free ribosomes (R) are also shown. **(H)** Pinocytosis in an endocrine cell. The plasma membrane of this cell displays numerous small, smooth invaginations (arrows), characteristic of non-clathrin mediated internalization of material.

The Plasma Membrane

The plasma membrane encloses the cytoplasm of a cell and carries out multiple functions. It forms both a barrier to, and an interface with, the cellular environment. The plasma membrane is a selectively permeable barrier that, by regulating what enters and exits a cell, is a primary determinant of the composition of the cytoplasm. The plasma membrane is associated with sensing mechanisms that transduce environmental information into a cytoplasmic or nuclear response. The plasma membrane is involved in cell-cell and cell-ECM attachments, and also contains cell-specific molecules that help identify cells, thereby helping to establish the appropriate position and arrangement of each cell in the body.

Barrier Functions

The hydrophobic nature of the central region of the lipid bilayer serves as a barrier to charged or large hydrophilic molecules; thus, the lipid bilayer is impermeable to small ions (e.g., Na^+, K^+, Cl^-) and proteins. Only small, uncharged molecules (e.g., CO_2, H_2O), or molecules freely soluble in lipid (e.g., steroid hormones, dioxin) are able to pass directly through the lipid bilayer. In this way, the plasma membrane is selectively permeable. However, materials can be transported into and out of the cell by specific transport mechanisms carried out by the plasma membrane (*see* Transport Functions). The carbohydrate moieties of glycolipids and glycoproteins also serve as barriers by impeding the access of molecules to the surface of the plasma membrane, which can also serve to protect plasma membranes exposed to harsh environments (e.g., the stomach and intestinal lumen).

Transport Functions

Because the lipid bilayer is impermeable to most types of organic molecules, the cell must possess mechanisms to move materials between the cytoplasm and the external environment. Two approaches are used by the cell to move material into and out of the cytoplasm: 1) transport through the membrane, and 2) transport involving membrane flow.

Transport Through the Plasma Membrane

Transport through membranes is mediated by integral membrane proteins, which help conduct material past the hydrophobic lipid bilayer in a number of ways. Integral membrane proteins can form channels by associating to form pore-like structures in the membrane. Such channels allow for diffusion of molecules small enough to fit through them. This type of transport allows for the flow of molecules down their concentration gradient and an expenditure of energy is not needed if the channel is open. Thus, molecules can move through protein channels by passive diffusion. Examples include ion channels that allow for the passage of ions such as Na^+ and K^+, and the connexons in gap junctions, which allow for the passage of molecules <1000 daltons through the plasma membrane. Whether these channels are open or closed is tightly regulated in order to prevent the constant leakage of small molecules into or out of the cell.

Integral membrane proteins can also act as carriers that bind specific molecules and help them traverse the lipid bilayer. Binding of the appropriate molecule to carrier proteins results in a conformational change in carrier protein structure such that the ligand is conveyed across the membrane. This type of transport is also driven by the concentration gradient of the ligand, and does not require the expenditure of energy by the cell. An example of transport by this method of facilitated diffusion includes glucose transporters in the basolateral membranes of intestinal epithelial cells.

Cells must also transport molecules against their concentration gradients, and this type of transport is carried out by integral membrane proteins that act as pumps and requires the expenditure of energy. This is referred to as active transport and is an essential process in living cells. Examples include a number of different ion pumps, which keep the cytoplasm relatively low in Na^+ and high in K^+. Ion pumps are vital elements of the plasma membrane and it has been estimated that as much as one-third or more of the energy consumed by a living cell is used to actively transport Na^+ out of the cell. The concentration gradient of certain ions established by these membrane pumps can itself serve as a motive force for other transport mechanisms. For example, in addition to moving out of a cell by facilitated diffusion, glucose is actively transported into cells by integral membrane proteins that bind both glucose and Na^+. Because these transporters bind both Na^+ and glucose, the high concentration of Na^+ outside the cell relative to the cytoplasm drives the movement of both Na^+ and glucose into the cell, against the concentration gradient of glucose.

Acquisition of glucose from the small intestine is an elegant example of how active transport can be coupled with facilitated diffusion to move molecules past epithelia. The apical membrane of intestinal epithelial cells contain Na^+-coupled active transporters that move glucose against its concentration gradient to accumulate in the cytoplasm. Consequently, the concentration of glucose is higher in the cytoplasm of these cells than in the extracellular spaces underlying them, and carrier proteins in the basolateral membranes of these cells allow for the facilitated diffusion of glucose out of the cell (down its concentration gradient) and into the circulation. It can be seen from this example that directional transport of molecules past epithelial cells requires that integral membrane transport proteins occupy specific locations within the plasma membrane (i.e., either the apical or basolateral membrane). How transporters are organized within the plasma membrane is determined by specific targeting mechanisms acting in conjunction with cell junctions and the cytoskeleton.

Transport Involving Membrane Flow

In addition to movement of materials through membrane channels, carriers, and pumps, the plasma membrane mediates the transport of material into and out of cells by membrane flow. Internalization of extracellular material can occur by entrapment in membrane-bound vesicles that pinch off from the plasma membrane and are transported into the cytoplasm for processing. This process, called endocytosis, can be subdivided into a number of different categories based on the mechanics of how the formation of vesicles occurs at the plasma membrane, and includes the formation of clathrin-coated vesicles from coated pits, and the formation of nonclathrin-coated vesicles derived from structures called caveolae.

Clathrin-coated vesicles comprise a major pathway in which specific extracellular molecules are recognized and bound to the plasma membrane prior to internalization. This process involves membrane receptors, which are integral membrane proteins of the plasma membrane that recognize specific ligands. The best understood example of this process involves how cholesterol is taken up by cells. In the circulation, cholesterol is packaged in low-density lipoprotein (LDL) particles, which are small particles composed of protein and cholesterol esters. Specific LDL receptors are present in the plasma membrane that bind and anchor LDL particles to the surface of the cell. Occupied LDL receptors form clusters in the membrane that recruit adapter proteins and the cytoplasmic protein clathrin. Clathrin molecules assemble beneath receptor clusters to form a basketwork that deforms the plasma membrane into an invagination referred to as a coated pit. Continued assembly of the clathrin coating results in the continued invagination, pinching off and release of a membrane-bound coated vesicle containing LDL receptor and LDL cargo into the cytoplasm. Once the vesicle is formed, the clathrin coating is disassembled and the clathrin recycled to the plasma membrane to assist in the formation of more coated pits. The clathrin-free vesicle then fuses with a membrane-bound compartment called an endosome, which, in addition to receiving vesicles from the plasma membrane, also receives lysosomal vesicles filled with hydrolytic enzymes packaged by the Golgi apparatus. Membrane-bound structures containing a mixture of LDL particles and acid hydrolases then arise from the endosome to form mature lysosomes. During this process, LDL dissociates from the LDL receptor in the acidic endosomal environment, and vesicles enriched in LDL receptor pinch off from the endosome to be recycled back to the plasma membrane. Digestion of LDL particles occurs in the lysosome, followed by the release of cholesterol from the lysosome into the cytoplasm of the cell. This process of receptor-mediated endocytosis is used to concentrate and internalize a number of extracellular molecules. Other common features of receptor-mediated endocytosis include the recycling of both clathrin and the receptor; the fusion of internalized vesicles with endosomes; and the formation of lysosomes (digestive organelles) for material internalized by this route.

A second endocytotic pathway exists that does not involve clathrin, and may bypass delivery of internalized material to lysosomes. In this pathway, which appears to involve both receptor-mediated endocytosis as well as the nonspecific internalization of extracellular fluid, vesicles are created from clathrin-free invaginated membrane regions called caveolae. Caveolae are associated with specialized membrane domains with distinct phospholipid contents called lipid rafts. Invagination and formation of vesicles in these areas does not require clathrin, and the vesicles formed may be transported directly to the Golgi apparatus or endoplasmic reticulum instead of the endosomes and lysosomes. Presumably this route is used for material that would be damaged or degraded by exposure to lysosomal enzymes. Many cells display a constitutive formation and internalization of these vesicles in a process sometimes referred to as pinocytosis, or *cell drinking*.

Signaling Functions

The plasma membrane serves as the interface with the cell environment and possesses a number of mechanisms to detect and transduce specific extracellular signals. Integral membrane proteins that serve as signal receptors can be categorized into three broad classes: ion channel-linked receptors, G-protein-linked receptors and enzyme-linked receptors.

Ion channel-linked receptors undergo conformational changes upon ligand binding, opening a membrane channel permeable to small ions. Examples of this type of receptor include some types of neurotransmitter receptors. G-protein-linked receptors are integral membrane proteins that, upon ligand binding, activate small GTP-binding proteins (G-proteins), which in turn activate other effector molecules, including ion channel-linked receptors and various enzymes (e.g., adenylate cyclase). Thus, G-protein-linked receptor activity can lead to ion transients across the plasma membrane, or the generation of second messengers such as cAMP. Examples of G-protein-linked receptors include polypeptide hormone receptors. The third class of membrane-associated signaling molecules are enzyme-linked receptors, which upon ligand binding activate an enzyme activity, which is usually a protein kinase or a guanylyl cyclase. Examples of enzyme-linked receptors are growth factor receptors, whose tyrosine kinase activity is an important regulator of the cell cycle.

In addition to membrane receptors involved in signal transduction events, cells possess other types of receptors not associated with the plasma membrane. For example, steroid hormones (e.g., estrogen and testosterone) are lipid soluble and pass directly through the lipid bilayer without the need to bind to proteins exposed on the external face of the plasma membrane. Receptors for these types of signaling molecules are found in the cytoplasm and nucleus.

Cell Junctions

Specializations of the plasma membrane also help cells adhere to each other and to the ECM. They form barriers against the diffusion of material between cells of an epithelium, and form channels between adjacent cells. All of these functions are carried out by cell junctions.

Cell-Cell Barrier Junctions

It is important for the body to prevent the passage of material between epithelial cells. For example, a major function of intestinal and bladder epithelia is to prevent the direct passage of the contents of the intestine and bladder into the body cavity. The ability of epithelia to form effective barriers between different compartments is owing to the presence of special cell-cell junctions called occluding, or tight, junctions. Tight junctions are composed of linear arrays of the integral membrane proteins occludin and claudin; forming a barrier between cells as strands of occludin encircling the apical part of adjacent epithelial cells line up and bind to each other. Epithelia that form strong barriers to intercellular leakage contain many tight junction strands, whereas more leaky epithelia generally display fewer strands. Occludin and claudin molecules are associated with other cytoplasmic proteins that appear to furnish some linkages with the cytoskeleton. Extensive cytoskeletal interactions, such as those associated with adhesive-type cell junctions, are not readily apparent in tight junctions.

Cell-Cell Adhesive Junctions

These specializations of the plasma membrane allow cells in an epithelium to bind tightly to each other, and can be subdivided into two categories: adherens junctions and desmosomes. Adherens-type cell-cell junctions form belt-like arrays encircling the apical part of epithelial cells and are associated with a thick band of microfilaments. Zonulae adherens are comprised of integral membrane proteins belonging to the cadherin family of proteins, a number of the linking proteins that interconnect cadherins to microfilaments, including vinculin and catenin, and associated microfilaments. Cadherins from adjacent cells bind tightly to each other in the presence of Ca^{2+}, and the chelate of Ca^{2+} promotes cell dissociation. Interestingly, β-catenin functions not only as an adherens plaque protein, but also as a transcription factor in the nucleus.

Thus, β-catenin may serve as a sensing device that translates changes in cell-cell adhesion into changes in gene activity.

Desmosomes are punctate cell-cell adhesive junctions that are associated with the intermediate filament cytoskeleton. Like adherens junctions, they are also composed of cadherin integral membrane proteins and proteins that interlink cadherins and the associated cytoskeletal filament system. One type of desmosomal cadherin is a protein called desmoglein, and the major linking protein of desmosomes is a member of the plakin family of proteins called desmoplakin. Desmosomal intermediate filaments form dense bundles that interconnect desmosomal plaques, thus strengthening cell-cell attachments that contribute to the mechanical integrity the epithelium.

Cell-Cell Communicating Junctions

Two types of cell-cell junctions, gap junctions and synapses, constitute specializations of the plasma membrane that allow cells to communicate with each other. Gap junctions are punctate structures that electrically couple cells through channels that provide for direct cytoplasmic communication between adjacent cells. Gap junctions are composed of clusters of pore-like structures, called connexons, that span the lipid bilayer and allow passage of molecules smaller than 1000 daltons. Connexons, made up of hexameric arrays of the integral membrane protein connexin, line up between adjacent cells to form continuous, tightly sealed channels from cell to cell. These connections maintain a barrier against leakage of material to, or from the extracellular compartment, but ions and small molecules can diffuse from cell-to-cell to permit electrical coupling. The electrical coupling of gap junctions perform vital functions in propagating the contraction of cardiac muscle. The conformation of connexons is regulated by Ca^{2+} such that they remain open in low concentrations of Ca^{2+}, but close in the presence of higher concentrations Ca^{2+}.

In addition to the direct coupling of cells at gap junctions, neurons also communicate with each other at synapses. At synapses, cells release neurotransmitters in quantal fashion by the regulated exocytosis of membrane-bound vesicles. Neurotransmitters rapidly diffuse across a narrow extracellular space to bind to specific receptors on the plasma membrane of the adjacent cell. These receptors are either ion channels, or, in some cases, G-protein-linked receptors. When stimulated by a neurotransmitter, ion channels open, allowing Na^+ to enter the cell by diffusing down its concentration gradient, thereby depolarizing the plasma membrane. G-protein-linked receptors that bind neurotransmitter release activated G-proteins that may subsequently activate and open other ion channels. Depolarization of the plasma membrane is conducted down the body of the stimulated cell, and can trigger the release of the neurotransmitter at synaptic junctions between the stimulated cell and other adjacent cells. In this way, signaling activity is propagated between cells interconnected by synapses.

Cell-ECM Adhesive Junctions

Cells elaborate two types of cell-ECM junctions that assist in their adherence to the ECM. Hemidesmosomes anchor epithelial cells to underlying connective tissue and are associated with intermediate filament cytoskeletal fibers, whereas focal contacts can be formed by many types of cells and involve microfilament-associated linkages with the extracellular matrix. Hemidesmosomes resemble half-desmosomes where proteins called integrins form the integral membrane component, linked to intermediate filaments by members of the plakin family of proteins. Integrins are also the integral membrane proteins of focal contacts, and are connected to bundles of microfilaments by vinculin, talin, and other linking proteins. Focal contacts are associated with protein kinases, called focal adhesion kinases (FAKs). FAKs are thought to help transmit the status of cell-ECM linkage at focal contacts to the cytoplasm and nucleus. Normal (noncancerous) cells must usually be in contact with a substrate to divide, and FAKs may be involved in relaying contact information to regulatory elements of the cell cycle.

Cell Protection and Cell Identity

Many proteins and lipids of the plasma membrane possess covalently-linked sugar groups. These sugar groups are asymmetrically distributed, and are presented exclusively on the ECM side of the membrane. Plasma membrane glycoproteins and glycolipids serve important roles in both protecting the membrane, and in identifying specific cell types. Epithelial cells lining the small intestine elaborate a thick glycocalyx that helps to protect them against the harsh digestive conditions of the intestinal lumen. Examples of cell recognition processes involving glycoproteins and glycolipids include the patterns of antigens on red blood cells responsible for blood groupings. Another example involves the initial adhesion of neutrophils to capillary endothelium in areas of inflammation. During inflammation, endothelial cells express the integral membrane protein P-selectin, which contains a lectin domain that recognizes a four sugar group (N-acetylglucosamine, galactose, fucose, and sialic acid) present on glycoproteins and glycolipids of neutrophils. Neutrophils adhere to endothelial cells expressing P-selectin. This facilitates their subsequent migration past the capillary bed to reach the sites of inflammation.

Endoplasmic Reticulum

The ER is a prominent organelle in most cells, and its total membrane area can constitute more than half of all the membrane of a cell. ER membranes delimit enclosed spaces that vary in shape from flattened sheets, or cisternae, to branching tubules, to distended sacs; the total enclosed lumenal space can occupy 10% or more of the cell volume. A number of distinct functions are carried out by the ER. The ER is the primary site of synthesis of membrane lipids and integral membrane proteins for all membranous organelles (ER, Golgi, lysosomes, endosomes, secretory vesicles, and plasma membrane) except mitochondria and peroxisomes. It is also the site of production of secreted proteins and lumenal proteins of ER, Golgi, and lysosomes. In addition, the ER functions in lipid synthesis, detoxification reactions, and Ca^{2+} regulation.

ER can be categorized as either rough (RER) or smooth (SER). RER is so designated owing to the presence of numerous ribosomes bound to the cytoplasmic surface of the cisternal membranes, whereas SER lacks associated ribosomes. These different forms of ER are specialized for different functions; RER is the site where integral, lumenal, and secretory proteins are synthesized, whereas SER is the major site of detoxification and lipid synthesis.

Protein Synthesis

Protein synthesis takes place on ribosomes, which are either located free in the cytoplasm, or attached to membranes of the ER (forming RER). Cytoplasmic proteins (e.g., cytoskeletal proteins) are synthesized by free ribosomes, whereas proteins associated with membranes (including the plasma membrane) or the lumenal compartments of membrane-bound organelles, as well as proteins destined for secretion, are synthesized by RER. The lipid components of membranes are also made by the ER, and both protein and lipid are delivered to the plasma membrane and most organelles by membrane flow. This involves the transport and fusion of membrane-bound vesicles between ER, the Golgi apparatus, and other target organelles. Exceptions to this pattern of membrane biogenesis and renewal include mitochondria and peroxisomes. Interestingly, most proteins of mitochondria, and all peroxisome proteins, are made by free ribosomes and subsequently delivered to these organelles via transport mechanisms that move individual proteins into or past their membranes. Membrane lipids are delivered to these organelles by transport proteins that extract lipid from ER membranes and insert them in the membranes of mitochondria and peroxisomes.

Whether ribosomes remain free in the cytoplasm or are bound to ER is determined by the amino acid sequence of the polypeptide chain as it emerges from the ribosome. ER-associated proteins possess a signal sequence that functions in docking the polypeptide to the membranes of the ER. The signal sequence is recognized and bound by a signal recognition particle, or SRP. The SRP in turn binds to a SRP receptor in the membrane of the ER. A protein translocator apparatus forms a pore in the ER membrane through which growing polypeptide chains can pass. It also associates with the SRP and SRP receptor, and receives the protein as translation proceeds. Thus, proteins with signal sequences are injected directly into the membrane of the ER as they are synthesized. The signal sequence, which is hydrophobic, remains inserted into the lipid bilayer while the rest of the protein spools past as it elongates. The relative orientation of the signal sequence influences whether the N- or C-terminus of the polypeptide is threaded into the ER lumen. Soluble lumenal proteins spool all the way through the bilayer, and a signal peptidase then clips the protein at the signal sequence, liberating the protein into the lumen. Single- and multi-pass membrane proteins are thought to achieve their conformations by internal stop-transfer and start-transfer sequences, which interact with the bilayer to either promote the passage of the growing polypeptide chain through the bilayer (start-transfer sequences), or halt the transmembrane passage (stop-transfer sequences). Multiple start- and stop-transfer sequences therefore can result in a polypeptide chain that doubles back and forth to penetrate the bilayer at multiple points, forming loops in both the cytoplasm and ER lumen.

A number of post-translational modifications of proteins occur in the ER, as well as in the Golgi apparatus. Chaperone proteins that help direct the correct folding of newly synthesized protein are present in the ER cisternae; disulfide bonds form and many proteins are glycosylated, or may have glycolipid anchors added. Glycosylation is carried out by the initial assembly of sugars into polymeric structures attached to the membrane lipid dolichol. The assembled carbohydrate group is then transferred from dolichol to the protein. The glycoprotein may be processed in the ER by trimming some sugars, and addition of others. Further processing of glycoproteins, and the formation of glycolipids, are major functions of the Golgi apparatus.

Lipid Synthesis

In addition to synthesis of proteins for membranes, lysosomes, and secretory vesicles, the ER is also responsible for the synthesis of most membrane lipid for all organelles (including mitochondria and peroxisomes). Enzymes involved in lipid synthesis are embedded in the membrane of the ER, with their active sites facing the cytoplasm. Fatty acids are added to glycerol phosphate to form phosphatidic acid, which then receives various head groups. Phosphatidylcholine, phosphatidylserine, and phosphatidylethanolamine are formed in this way and initially added to the cytoplasmic leaflet of the ER lipid bilayer. Phospholipid translocators (*flippases*) are present in the membrane that transfer choline-containing phospholipids from the cytoplasmic half to the lumenal half of the bilayer. These translocators keep the total numbers of phospholipid molecules approximately equal between the two layers, but give rise to membrane asymmetry in the distribution of these lipids. Sphingomyelin synthesis is more complex; serine is first attached to fatty acids to form ceramide, which is exported to the Golgi apparatus, where phosphocholine head groups are added. Mitochondria and peroxisomes appear to receive their membrane lipid from the ER through the activity of phospholipid exchange proteins, which transfer phospholipids between membrane systems by extraction and insertion of individual lipid molecules. In addition to membrane lipid synthesis, the ER plays important roles in other aspects of lipid metabolism. For example, steroid hormones are synthesized from cholesterol in the SER.

Detoxification

Harmful substances that are relatively insoluble are difficult to clear from the cell. Such substances can occur as either environmental contaminants or as products of metabolism. SER contain a variety of enzymes that are able to process insoluble toxicants to make them more water-soluble and amenable for excretion. The best studied detoxification enzymes are members of the cytochrome P450 family of enzymes. Liver hepatocytes are among the most active cells involved in detoxification reactions, and contain large amounts of SER that house the P450 enzymes. The quantity of SER within a cell can fluctuate in response to different levels of exposure to toxic compounds.

Ca^{2+} Regulation

The ER membrane contains Ca^{2+}-ATPases that actively transport cytoplasmic Ca^{2+} into the ER lumen. This activity keeps cytoplasmic Ca^{2+} levels very low, which is necessary to allow Ca^{2+} to effectively function as a signaling molecule. In electrically excitable cells, depolarization of the plasma membrane promotes influx of Ca^{2+} from outside the cell; in nonexcitable cells, however, most of the Ca^{2+} released into the cytoplasm comes from the ER. ER membranes contain Ca^{2+} release channels that are activated by inositol triphosphate (IP3), a signaling molecule released by the activation of certain G-protein-linked receptor proteins at the plasma membrane. The contraction of muscle cells is triggered by Ca^{2+}, and these cells possess an extensive and specialized SER system, the sarcoplasmic reticulum, that contains a second type of Ca^{2+}-release channel in the SER membrane. After release from the ER, Ca^{2+} concentrations are lowered in the cytoplasm by the activity of plasma membrane and ER pumps.

Golgi Apparatus

The Golgi apparatus functions as the post office of a cell, packaging and directing different types of cargo from the ER to different organelles and the plasma membrane. In addition to packaging and targeting membrane-associated protein and lipid, as well as secreted protein, to their appropriate destinations, the Golgi apparatus modifies certain proteins and lipid received from the ER. Glycolipids are formed in the Golgi by the addition of oligosaccharide chains to ceramide; in addition, further processing of glycoproteins continues in the Golgi.

The Golgi apparatus is made up of a set of flattened, membrane-bound cisternae and associated tubulovesicular elements and membrane-bound vesicles in the process of being transported between ER, Golgi, and other locations. The stacks of Golgi cisternae are biochemically distinct, and the entire stack is polarized, so that a *cis*, or entry face, and a *trans*, or exit face, exist. The *cis* face lies adjacent to ER, and is the site of vesicular traffic back and forth between the ER and Golgi. The *trans* face is the site of formation of a number of types of vesicles that convey material to the plasma membrane, produce secretory vesicles, and form lysosomes. Integral membrane proteins, membrane lipids and soluble cisternal protein formed by the ER traverse the Golgi and are targeted to their appropriate destinations by this organelle. Three major routes of export from the Golgi occur: 1) constitutive delivery of membrane-bound vesicles to the plasma membrane; 2) formation of secretory vesicles whose exocytosis is regulated; and 3) formation of lysosomes. Specific targeting signals are associated with the formation of lysosomal vesicles and secretory vesicles; however, the pathway from the Golgi apparatus to the plasma membrane appears to be largely constitutive and unregulated, forming a default pathway.

Proteins destined to be secreted in a regulated manner (e.g. release of hormones from endocrine cells) are concentrated and packaged in membrane-bound vesicles formed by the *trans* Golgi. These secretory vesicles are stored in the cytoplasm until signals to fuse with the plasma membrane are received, resulting in the liberation of their contents outside the cell. Targeting mechanisms exist that direct secretory vesicles to the appropriate cellular location for release. For example, some secretory vesicles are released from the apical plasma membranes of epithelial cells, whereas others fuse with basolateral membranes.

Proteins destined for lysosomes are tagged with mannose-6-phosphate (M6P) groups in the Golgi. M6P receptors are present in Golgi membranes, and vesicles containing lysosomal proteins bound to M6P receptors bud off from the Golgi and fuse with endosomes to form mature lysosomes. During this process, M6P receptors are recycled for repeated use by vesicular trafficking from endosome to Golgi apparatus.

Lysosomes

Lysosomes are the digestive organelles of the cell, and are filled with acid hydrolases that are most active at a pH of about 5.0. Lysosomal vesicles from the Golgi apparatus fuse with endosomes that have received material from endocytotic vesicles. Endosomes have a moderately acidic pH of about 6.0 that promotes dissociation of ligand from internalized plasma membrane receptors as well as dissociation of lysosomal hydrolases from M6P receptors. Both types of receptors are recycled by being routed back toward the plasma membrane and the Golgi apparatus in membranous vesicles that are pinched off from endosomes. The endosome then matures to form a lysosome by condensing into a spherical or irregular membrane-bound structure. Proton pumps in the membrane of the maturing lysosome lower the pH inside the organelle to maximally activate the acid hydrolases to digest the internalized material. Other transporters exist in the lysosomal membrane to allow digested organic molecules to enter the cytoplasm for use by the cell.

In addition to the confluence of lysosomal and endocytotic vesicles at endosomes, material can be targeted for lysosomal degradation by at least two other mechanisms. Neutrophils and macrophages are cells specialized for the engulfment of bacteria and other large particulate material, which are internalized by phagocytosis. Lysosomes fuse with these large phagocytotic vesicles to deliver their hydrolases, resulting in the formation of phagosomes. Lysosomes also contribute to the breakdown of cellular material that is not needed or should be turned over. Excess, old, or malfunctioning organelles can be targeted for destruction by becoming enveloped by cisternae of ER, which then fuse with lysosomal vesicles to form autophagosomes. Recently, evidence has been gathered suggesting that a fourth route to lysosomes may exist that involves the transport of single cytoplasmic molecules through the lysosomal membrane by specific transport proteins.

Membrane Flow

It is apparent that there is a complex, but effective and elegant, flow of membranes and molecules between the various organelles and cytoplasmic compartments of eukaryotic cells. With the exception of mitochondria and peroxisomes, the membranes of all organelles, vesicles, and the plasma membrane are initially produced by the ER, modified and packaged by the ER and Golgi, then targeted and delivered via the trafficking of membrane-bound vesicles. A number of different types of signals and targeting mechanisms are used to regulate this flow of the membrane. Between the ER and Golgi, vesicles are coated with coat proteins (COPs), that either direct vesicles from the ER to the Golgi (COPII), or direct vesicles from the Golgi to the ER (COPI). Although no mechanisms appear to exist that restrict the flow of material from the ER to the Golgi, the amino acid sequences KDEL and KKXX (where X is any amino acid) mark lumenal and integral membrane proteins, respectively, for return transport from the Golgi to the ER. Thus, proteins with these sequences are essentially restricted to the ER by being rapidly returned from the Golgi apparatus. The exact mechanism of membrane flow through the Golgi apparatus has been a matter of some debate, but at least part of the flow appears to be carried out by membrane-bound vesicles. The flow of membrane from the Golgi to plasma membrane includes a constitutive, default pathway that operates in the absence of specific targeting signals. However, delivery of material to secretory vesicles and lysosomes requires defined targeting information. Interestingly, clathrin is involved in the formation of lysosomal and secretory vesicles from the *trans* Golgi, but not in the constitutive formation of vesicles destined for the plasma membrane. In addition to the ER and Golgi apparatus, bi-directional membrane flow also occurs between plasma membrane and endosome, and plasma membrane and Golgi. Clathrin-coated endocytotic vesicles from the plasma membrane travel inward to fuse with endosomes, and the receptors subsequently return to the plasma membrane via small vesicles formed from endosomal membranes. Endocytotic vesicles formed from caveolae may bypass lysosomes and fuse directly with Golgi or ER. Although the details of vesicular targeting are not well understood, it has been proposed that SNARE (soluble N-ethylmaleimide-sensitive-factor attachment protein receptors) proteins on the surfaces of membrane-bound vesicles (v-SNAREs) and target organelles (t-SNAREs) mediate the correct patterns of docking between vesicles and organelles.

Mitochondria

The primary function of mitochondria is to convert energy sources into forms that can be used to drive cellular reactions. Not surprisingly, they comprise a significant volume of the cell—normally, about 20% of the total cell mass. Mitochondria replicate by a process involving growth and fission of pre-existing mitochondria. Interestingly, mitochondria contain their own DNA that resembles the genome of prokaryotes. Based on this and other lines of evidence, it appears that mitochondria (and plastids in plant cells) arose by the colonization of eukaryotic cells by prokaryotes early in their evolution. Although mitochondria contain DNA and are able to carry out transcription and translation, they only produce about 5% of their protein, the rest being encoded by nuclear genes and synthesized by cytoplasmic ribosomes. They also appear to obtain most of their membrane lipid from the ER, mediated by of phospholipid transfer proteins that shuttle these molecules from the ER to the various organelles, including mitochondria.

Cells use ATP as their primary energy source, and the main function of mitochondria is the production of ATP from food sources. Energy is obtained from the oxidation of food by the sequential transfer of high energy electrons to lower energy states; the released energy is used to drive membrane-bound proton pumps, thus establishing an electrochemical gradient. Protons are then allowed to flow back across the membrane down their concentration gradient, and the released energy is used to drive the synthesis of ATP from ADP and Pi. The electrons are ultimately transferred to O_2, and the entire process is therefore referred to as oxidative phosphorylation.

Mitochondrial Structure

Mitochondria are composed of two membranes, that enclose distinct compartments. The outer mitochondrial membrane is somewhat permeable to small molecular weight compounds (<5000 daltons); conversely, the inner membrane contains a very high ratio of protein to lipid, and movement of material past this membrane is tightly regulated. The space between the two membranes is called the *intermembrane space*, and the compartment delimited by the inner membrane is called the mitochondrial matrix. The inner membrane is folded into sheet- or tube-like invaginations within the matrix, thus increasing its surface area. The increased surface area of this membrane allows mitochondria to house greater numbers of electron transport enzyme systems and ATP synthase complexes. The intermembrane space resembles the cytoplasm, but the internal matrix is biochemically distinct. The matrix is the site of conversion of pyruvate and fatty acids to acetyl CoA and is the location of the citric acid cycle, where acetyl CoA is oxidized.

Chemiosmotic Generation of ATP

With respect to energy production, the pathways of carbohydrate and lipid metabolism converge in the generation of acetyl CoA in the mitochondrial matrix. Carbohydrate is converted to glucose-6-phosphate, which, as a substrate for glycolysis, gives rise to two pyruvate molecules. Pyruvate is transported to the mitochondrial matrix were it is converted to acetyl CoA by the pyruvate dehydrogenase complex. Fatty acids are oxidized in the mitochondrial matrix, releasing acetyl groups that are then linked to CoA. Acetyl CoA derived from carbohydrate and fatty acid metabolism is fed into the citric acid cycle, resulting in the production of CO_2 and NADH. NADH conveys high energy electrons to the electron transport chain, which is located on inner mitochondrial membrane. The electron transport chain is a complex composed of at least 40 different proteins. The actual transfer of electrons is carried out by a number of different heme groups linked to various cytochromes, iron-sulfur center-containing proteins, ubiquinone, copper atoms, and a flavin. These are organized into three large enzyme complexes, with ubiquinone and cytochrome C serving as carriers of electrons between the complexes. The three complexes are the NADH dehydrogenase complex, the cytochrome B-C1 complex, and the cytochrome oxidase complex. Electrons shuttled across these complexes move from high to low energy states, with the released energy used to pump H^+ from the matrix to the intermembrane space. Then, H^+ is allowed to flow down its concentration gradient (from the intermembrane space to the matrix) through the ATP synthase complex. This is a large transmembrane complex of about 500,000 daltons that contains multiple proteins, and constitutes about 15% of the total inner membrane protein. The energy that is released is used to couple Pi to ADP to make ATP. Finally, the electrons used to drive the H^+ pumps are combined with O_2 and H^+. Thus, the generation of ATP from high energy electrons by this chemiosmotic mechanism consumes O_2 and produces water.

Other Mitochondrial Functions

In addition to converting food energy into ATP, mitochondria also are involved in a number of other functions, including Ca^{2+} regulation and apoptotic signaling. Like the ER, mitochondria sequester Ca^{2+} to help keep cytoplasmic levels low. In addition to producing ATP, the H^+ gradient can be used to import Ca^{2+} into the mitochondrial matrix, where it is used in part to help regulate the activity of certain mitochondrial enzymes. Deposits of calcium can be formed in mitochondria in response to high cytoplasmic levels of Ca^{2+}. Mitochondria are also involved in the regulation of programmed cell death, or apoptosis. A central mechanism by which apoptosis is carried out involves activation of the caspase family of proteases. Release of cytochrome C by mitochondria facilitates caspase activation by forming a complex with other molecules (e.g., APAF-1) and pro-caspases to activate the caspase cascade. Release of cytochrome C from mitochondria can occur in response to specific membrane signaling events, as well as from cytoplasmic, mitochondrial, or nuclear damage.

Peroxisomes

Peroxisomes are membrane-bound vesicular organelles that are involved in various oxidative reactions. They contain high concentrations of enzymes that are able to form H_2O_2 by the transfer of hydrogen atoms from substrates to molecular oxygen. In addition, peroxisomes contain catalase, which breaks down H_2O_2 to oxidize various substrates, including some types of toxins. Like mitochondria, peroxisomes replicate by fission and growth of pre-existing organelles. All protein and lipid of the peroxisome is synthesized in the cytoplasm, and subsequently imported into the peroxisomal membrane and lumen.

The Cytoskeleton

The cytoskeleton of eukaryotic cells is composed of a complex system of proteinaceous filaments that are present in both cytoplasm and nucleus. The three major cytoskeletal systems elaborated by cells are microfilaments, microtubules, and intermediate filaments. These different cytoskeletal systems are biochemically and functionally distinct.

Microfilaments

Microfilaments are small solid filaments about 6 nm in diameter, composed of the 45 kDa globular protein actin. Actin is one of the most abundant proteins in cells, composing up to 5% or more of the total cell protein. Microfilaments help support and organize the plasma membrane and are involved in cell motility and the maintenance of cell shape, serving as the *muscle* of the cell.

A large number of actin-associated proteins mediate the functions of microfilaments, including regulating actin polymerization (e.g., profilin, WASp, and ARP), cross-linking microfilaments to form organized arrays (e.g., filamin, fimbrin, and villin), interacting with membrane proteins to establish and maintain distinct membrane domains (e.g., vinculin, catenins, and Z0-1 of tight junctions), and functioning as motor proteins (e.g., type I and II myosins) to carry out motility events. Much of the actin in cells is present as soluble monomers (g-actin) bound to profilin. This interaction inhibits the polymerization actin monomers into filaments (f-actin). Signals from the plasma membrane, mediated in large part by small GTP-binding proteins (e.g., rac and rho), activate WASp and ARP proteins, to promote the dissociation of profilin from g-actin and seed the growth of new microfilaments from the sides of pre-existing microfilaments. Microfilament polymerization is controlled by the addition of capping proteins to the end of growing filaments, and turnover of filaments occurs by the actions of microfilament cutting proteins such as gelsolin, followed by depolymerization of f-actin to g-actin and association of the latter with profilin.

Although some cell movements and membrane activities appear to be driven by the polymerization and depolymerization of actin, many other types of actin-associated motility require the interaction of microfilaments with myosin motor molecules. Myosin functions as an actin-activated ATPase, undergoing cycles of microfilament attachment and detachment, with associated conformational changes, resulting in *power strokes*. These events are linked to the binding, hydrolysis, and release of ATP, Pi, and ADP. Myosin activity can slide microfilaments past each other, transport vesicles and other cargo down microfilaments and deform membranes that are attached to microfilaments.

A number of mutations are known that interfere with microfilament functioning. WASp protein was discovered as the protein mutated in Wiscott-Aldrich syndrome, which results from deficits in the ability of actin to polymerize. Dystrophin is a large linking molecule that interconnects submembrane arrays of microfilaments to integral membrane proteins and ECM proteins in skeletal muscle cells. Mutations in dystrophin that interfere with its ability to link microfilaments with the plasma membrane weaken the plasma membrane, causing the eventual death of the muscle cell and giving rise to some forms of muscular dystrophy.

Microtubules

Microtubules are small hollow proteinaceous tubes about 25 nm in diameter, composed of the protein tubulin. Microtubules function to organize the cytoplasm and mediate intracellular motility events. They are associated with motor proteins that interact with membrane-bound organelles and vesicles to help determine their location and organization within the cytoplasm. They also carry out crucial functions in cell division, forming the spindle apparatus that segregates replicated chromosomes among the daughter cells. Microtubules also support and power cilia and flagella, which are highly motile appendages produced by ciliated epithelial cells and sperm.

Unlike microfilaments and intermediate filaments discussed below, microtubules are associated with a distinct organizing center, called the centrosome (or MTOC, for microtubule-organizing center). The centrosome is composed of a pair of centrioles surrounded by an amorphous mass of pericentriolar material. Centrioles themselves are short, barrel-like arrays of microtubules and are associated with the ability of the centrosome to replicate. Pericentriolar material is a biochemically complex layer of amorphous material that surrounds the centrioles, which both nucleates microtubule growth and anchors the ends of microtubules. Three types of tubulin genes are expressed in eukaryotic cells, including α-, β-, and γ-tubulin. Microtubules are composed of heterodimers of α- and β-tubulin; γ -tubulin is a component of the pericentriolar material. Microtubules possess an intrinsic polarity and are all oriented so one end (the *minus* end) is anchored in the pericentriolar material, with the free end (the *plus* end) extending into the cytoplasm. Microtubule polymerization and depolymerization occurs at the plus end and involves the addition or removal of α-β heterodimers. Heterodimers of α- and β-tubulin are associated with GTP or GDP. GTP-containing heterodimers readily polymerize, whereas GDP-containing heterodimers bind much more weakly to each other and tend to dissociate. GTPases that hydrolyze microtubule-bound GTP to GDP are present in the cytoplasm; however, because heterodimer addition and removal occur at just the plus end, as long as the terminal tubulin subunits are associated with GTP, the microtubule will grow by the addition of GTP-containing heterodimers. Occasionally, the GTPase activity catches up with a growing end of the microtubule, hydrolyzing GTP to GDP in the terminal subunits. At this point, the microtubule rapidly disassembles, shrinking in size back toward the centrosome. Depolymerizing microtubules can be *rescued* and re-grown if sufficiently high concentrations of GTP-containing tubulin heterodimers are present so that the GTP *cap* can be re-established. Because of these events, most microtubules in the cell continuously oscillate between slow growth and rapid depolymerization, a feature that has been called *dynamic instability*.

Like the microfilament cytoskeleton, the organization and functions of microtubules are regulated and carried out by associated proteins. Microtubule-associated proteins are generally categorized as either structural proteins or motor proteins. Structural proteins include higher molecular weight proteins called MAPs (for microtubule-associated protein), and lower molecular weight tau proteins. Structural MAPs and tau are thought to help organize microtubule arrays in the cytoplasm. Microtubule-associated motor proteins include dynein and kinesin, both of which, like myosin, undergo cycles of binding, conformational changes, and dissociation in an ATP-dependent manner to move microtubules past each other, or to move cargo along microtubules. Microtubule-mediated intracellular transport is carried out by multiple members of both the dynein and kinesin families of protein, but ciliary and flagellar motility selectively utilize dynein. The microtubule bundle, or axoneme, supporting a cilium is anchored in a specialized centrosome called a basal body. Unlike regular centrosomes, axoneme microtubules originate as direct extensions from one of the centrioles in a basal body, and not from associated peri-centriolar material. Axoneme microtubules form circular arrays of doublets surrounding a central pair of microtubules. The outer microtubule doublets are associated with dynein, which spans adjacent microtubule pairs. Dynein motor activity attempts to slide microtubule pairs past each other, which is converted into a bending of the cilium because the bases of the microtubule pairs are attached to the basal body and are not free to slide. In this way, hydrolysis of ATP by dynein powers the rapid, whip-like movements of cilia and flagellae in a microtubule-dependent manner. Other forms of microtubule-mediated motility occur in the cytoplasm, where membrane-bound vesicles, organelles, and other cargo associated with dynein or kinesin move along microtubules. Motor proteins exhibit a directionality which allows for vectorial transport of material within the cell. Dyneins move cargo toward the minus ends of microtubules and kinesins generally move cargo toward the plus ends of microtubules (although minus end-directed members of the kinesin family are known).

In dividing cells, centrosomes replicate along with DNA in S-phase and subsequently participate in the formation of the mitotic spindle. Daughter centrosomes move apart and promote the complete reorganization of the microtubular cytoskeleton prior to, and during, nuclear envelope breakdown. The plus ends of microtubules radiating from the centrosomes, now called spindle poles, bind and become stabilized by the kinetochores of chromosomes, forming distinct bundles of kinetochore microtubules. Microtubule-mediated motility events at the kinetochore line chromosomes up on the metaphase plate and are subsequently responsible for the separation of daughter chromosomes in anaphase of mitosis. Pinching the cell into two daughter cells (mediated by bundles of microfilaments in association with the plasma membrane at the cleavage furrow) leads to the inheritance of the correct number of chromosomes and a single centrosome by each daughter cell. Interestingly, other organelles such as mitochondria and peroxisomes appear to be randomly apportioned to each daughter cell by virtue of the fact that they are distributed throughout the cytoplasm, whereas the ER and Golgi apparatus vesiculate and disperse throughout the cytoplasm early in mitosis to be inherited in the same way.

A number of drugs interfere with microtubule dynamics; examples include colchicine, which actively promotes tubulin depolymerization; and taxol, which stabilizes microtubules and inhibits depolymerization. Both of these drugs are toxic to cells, indicating that the oscillation between polymerization and depolymerization is crucial to microtubule function. A number of taxol-based compounds have been developed for use in the chemotherapeutic treatment of cancer, highlighting the importance of microtubule dynamics in cell division.

Intermediate Filaments

Intermediate filaments (IF) are solid filaments about 10 nm in diameter, made up of one or more of a large family of intermediate filament proteins. Intermediate filaments are found in both the cytoplasm and the nucleus. They function in strengthening the cytoplasm of cells, as well as in mechanically integrating cells of a tissue by interconnecting desmosomes and hemidesmosomes.

The IF family of proteins is the most complex family of cytoskeletal proteins, with over 50 different IF gene products elaborated by cells of higher vertebrates. IF proteins can be divided into five groups: 1) acidic keratins, 2) neutral/basic keratins, 3) vimentin-like proteins, 4) neurofilament proteins, and 5) lamins. Intermediate filament proteins are expressed in tissue-specific patterns, with epithelial cells containing keratins, cells of mesenchymal origin expressing vimentin-like IF proteins, and neuronal cells expressing neurofilament IF proteins. Lamins are present in essentially all nucleated cells and form a filamentous network underlying and supporting the inner membrane of the nuclear envelope. There is evidence that lamins help organize chromatin and are involved in some aspects of DNA synthesis, as well.

IF proteins are long, rod-like molecules that contain a central domain rich in α-helices. The rod domain of IF proteins coil around each other to form coiled-coil dimers, which then associate into higher order structures, much like weaving together individual strands to form a rope. The polymerization state of IF proteins is mediated by phosphorylation, and it has been suggested that hyperphosphorylation of IF proteins leads to dissociation of IFs by repulsion of subunits bearing multiple negative phosphate charges. One of the best studied examples is the depolymerization and repolymerization of nuclear lamina filaments during cell division. At the onset of mitosis, lamins are hyperphosphorylated and the nuclear lamina depolymerizes, facilitating nuclear envelope breakdown. At the end of mitosis, lamins are dephosphorylated and the nuclear lamina and nuclear envelope re-forms around the daughter nuclei.

Only a relatively few IF-associated proteins are known, and these appear to help organize intermediate filaments and mediate interactions with other cytoskeletal proteins and organelles. Intermediate filaments frequently form a dense basketwork around the nucleus; thus, the nucleus of many cells is supported both by an internal IF lamina, as well as protected by a cytoplasmic network of IF fibers.

Nuclear Organization

The largest and most prominent structure in most cells is the nucleus. It serves as the repository and organizing center for the genome. In humans, chromosomal DNA—of a total length of about two meters when fully extended—is packaged into an average nuclear size of about 10 μ*M* (1/100 of a millimeter) in diameter. The DNA must be packaged in such a way as to allow access to transcriptional and replication machinery, and extensive nuclear-cytoplasmic transport must occur. These functions are mediated by the organization of DNA within the nucleus, and the organization of the nuclear envelope.

The nucleus is bounded by the nuclear envelope, which is comprised of two membranes (the inner and outer nuclear membranes), the nuclear lamina, and numerous nuclear pores that span the inner and outer membranes. Nuclear pores are multimolecular arrays exhibiting eight-fold symmetry that are involved in the exchange of material between cytoplasm and nucleus. Material moves through nuclear pores by both passive diffusion and by active transport; molecules smaller than 5000 daltons are freely permeable between nucleus and cytoplasm, but those larger than about 60,000 daltons must be actively transported. Molecules between these sizes can move between nucleus and cytoplasm without being actively transported, but take longer to equilibrate with increasing size. Proteins actively transported into the nucleus contain nuclear localization sequences that are recognized by the pore complexes. Nuclear localization signals vary in amino acid sequence, but usually contain a number of lysines and are positively charged.

The first step in DNA packaging involves the winding of DNA around octamers of histone proteins, which are positively charged proteins that closely interact with the negatively charged DNA. This first order of DNA packing gives rise to a *beads on a string* appearance, with the beads, or nucleosomes, composed of about 200 base pairs of DNA wrapped almost twice around a histone octamer core. Nucleosomes are further coiled to form a 30 nm diameter solenoid fiber, which loops out to form euchromatin (loosely compacted and generally active DNA), or becomes more tightly compacted to form heterochromatin (tightly compacted and relatively inactive DNA). Each chromosome usually consists of a mixture of heterochromatin and euchromatin, and occupies a more-or-less defined region within the nucleus. Portions of a number of chromosomes that contain amplified sequences encoding ribosomal RNA and ribosomal protein mRNAs cluster together and associate with a number of other protein elements in the nucleus to form the *nucleolus*. This is a specialized area where transcription of ribosomal genes and assembly of ribosomal subunits occurs.

Mounting evidence suggests that chromosomes may be organized on a protein or protein-RNA based scaffolding. This scaffolding, or nuclear matrix, is biochemically ill-defined, but appears to be composed of filaments that form a three-dimensional meshwork within the nucleus, which is surrounded by the denser filamentous mat of lamin IFs underlying the nuclear envelope. Although the composition of the nuclear matrix is not well-understood, it is possible that lamin IFs are not restricted to the nuclear periphery, but contribute to at least some of the matrix fibers. Certain DNA sequences bind to the nuclear matrix much more tightly than others, leading to the proposal that distinct matrix attachment regions, or MARs, periodically link chromosomes to the matrix. These linkages result in the formation of large (20–200 kilobase) loops of DNA tethered to the matrix at MAR domains. MAR DNA sequences do not display a rigid consensus sequence, but have a number of distinguishing features, including being relatively AT-rich, histone-poor, and possessing multiple topoisomerase II binding sequences. MAR domain DNA also may confer position independent expression of exogenous DNA incorporated into random sites within the genome. Thus, it has been proposed that MAR domains form long-range regulatory elements helping to control gene expression.

The high concentrations of DNA, RNA and protein in the nucleus make it challenging to study nuclear structure. However, some structural aspects of gene expression and mRNA processing can be visualized by electron microscopy in the form of perichromatin fibers and interchromatin granules. These structures represent sites of RNA processing, and include mRNA as well as associated ribonucleoproteins and RNAs that form elements of the splicing machinery, including spliceosomes and splicing islands.

Conclusion

Living eukaryotic cells must carry out and coordinate an enormous number of biochemical reactions in order to obtain and convert energy to usable forms, break down and interconvert organic molecules to synthesize needed components, sense and respond to environmental and internal stimuli, regulate gene activity, sense and repair damage to structural and genomic elements, grow and reproduce. This level of complexity requires that biochemical reactions be highly organized and compartmentalized, and this is the major function of cell organelles and the cytoskeleton. Cells have elaborated an elegant cytoplasmic membrane system composed of the nuclear envelope, endoplasmic reticulum (ER), Golgi apparatus, and associated endocytotic, endosomal, lysosomal, and secretory vesicles. These membrane systems serve to both organize and compartmentalize biochemical reactions involved in protein and lipid synthesis, targeting, and secretion. The cytoskeleton facilitates not only cytosolic molecular interactions, but also serves to organize the entire cytoplasmic membrane system. The key to cellular life is organization, and eukaryotic cells display a remarkably rich and elegant architecture to carry out the demands of life.

Glossary and Abbreviations

Actin The protein used to form microfilaments. Actin can be either soluble (g-actin) or polymerize to form microfilaments (f-actin).

Adherens Junction A type of cell-cell adhesive junction in which bundles of microfilaments are connected to the plasma membrane via linking proteins (e.g., catenin). The linking proteins connect microfilaments to integral membrane proteins called cadherins, which bind to each other in the presence of Ca^{2+} to adhere cells together.

ADP Adenosine 5'-diphosphate. A nucleotide associated with cellular energy regulation. The release of one of the three high-energy phosphate groups of ATP yields usable energy and ADP (which contains two phosphate groups and is at a lower energy state). ATP can be regenerated from ADP with the input of energy to attach a third high-energy phosphate group.

Apoptosis The process by which cells actively destroy themselves. Specific biochemical pathways exist, that, when activated, result in the destruction of key cytoplasmic and nuclear proteins.

ATP Adenosine 5'-triphosphate. High energy molecule, most of which is normally generated by oxidative phosphorylation in mitochondria. ATP is the primary source of cellular energy used to power enzymatic reactions.

Basal body A specialized type of centrosome (or MTOC) that gives rise to a cilium or flagellum.

Cadherin An integral membrane protein found in desmosomes and adherens junctions. In the presence of Ca^{2+} , cadherins from adjacent cells bind, adhering cells to each other. Cadherins are connected to microfilaments at adherens junctions via linking proteins such as catenin, and to the intermediate filament cytoskeleton at desmosomes via linking proteins such as desmoplakin.

Caspases Proteases that are activated during apoptosis. Caspases destroy key cellular components, as well as activate nucleases, thus promoting nuclear disassembly and cell death.

Catenin A type of linking protein found in adherens-type cell-cell junctions. Catenins are particularly interesting in that they can translocate to the nucleus and function as transcription factors, thus transducing events at the cell surface into changes in nuclear activity.

Caveolae Invaginations of the plasma membrane involved in endocytosis. Caveolae are not associated with clathrin, but possess a distinct lipid makeup, and may constitute a specialized type of lipid raft.

Cell Junction Specializations of the plasma membrane that allow for anchorage and communication between cells, and between cells and the extracellular matrix.

Centriole A short, barrel-like structure composed of a cylindrical array of microtubule triplets. Centrioles are associated with centrosomes, basal bodies, and spindle poles, and function in the replication of these microtubule organizing centers.

Centrosome The organizing center for the microtubular cytoskeleton. Centrosomes are composed of centrioles and pericentriolar material. Three forms of centrosomes are found in cells, including the single centrosome of non-dividing cells, the two spindle poles of dividing cells, and the basal bodies of ciliated cells.

Cilia Motile, whip-like extensions of the cell, supported by a bundle of microtubules that are connected to basal bodies. Cilia actively beat back and forth as a result of interactions between microtubules and the motor protein dynein, in an ATP-requiring process.

cis-Golgi The portion of the Golgi apparatus that receives membrane-bound vesicles from the ER.

Clathrin A protein involved in receptor-mediated endocytosis. Clathrin forms a coat around invaginations of the plasma membrane, forming coated pits, which subsequently pinch off into the cytoplasm to form coated vesicles.

Claudin An integral membrane protein associated with tight junctions that helps form a transcellular barrier between cells.

Coated Pit *see* **Clathrin**

Coated Vesicle *see* **Clathrin**

COPs Proteins involved in directing membrane-bound vesicles between the ER and the Golgi apparatus.

Cytochrome P-450 A group of proteins involved in detoxification reactions.

Cytoskeleton A system of filaments and tubules in the cytoplasm and nucleus that perform numerous functions, including maintaining cell shape, driving cell motility and cell division, and organizing the cytoplasm. Three major types of fibers comprise the cytoskeleton: microfilaments, microtubules, and intermediate filaments.

Dalton A measure of the mass of a molecule; one dalton is about the mass of a hydrogen atom. Small proteins are a few thousand daltons in size, medium and large proteins range from tens of thousands to few hundred thousand daltons, respectively.

Desmosome A type of cell-cell adhesive junction in which bundles of intermediate filaments are connected to the plasma membrane via linking proteins (e.g., desmoplakin). The linking proteins connect intermediate filaments to integral membrane proteins called cadherins, which bind to each other in the presence of Ca^{2+} to adhere cells together. Numerous desmosomes are found in tissues subjected to mechanical stress, such as epidermis and heart muscle.

DNA Deoxyribonucleic acid.

ECM Extracellular matrix.

Endocytosis A process by which cells internalize material via the formation of depressions in the plasma membrane which pinch off in the cytoplasm to form membrane-bound vesicles containing the engulfed material. Specific extracellular molecules can be concentrated and internalized by this method in a process called receptor-mediated endocytosis.

Endoplasmic Reticulum An extensive, membrane-bound cytoplasmic organelle involved in protein and lipid synthesis, as well as in detoxification reactions and Ca^{2+} regulation. Membranes of the ER form an enclosed compartment that range in morphology from flattened sheets to an interconnected tubular network. Protein and lipid products made by the ER can be delivered to other parts of the cell or secreted, via membrane-bound vesicles that traffic between the ER, Golgi apparatus, and plasma membrane. Protein synthesis occurs in ER that possesses attached ribosomes (RER), and lipid synthesis is primarily associated with ER lacking ribosomes (SER).

Endosome A membrane-bound structure formed by the coalescence of endocytotic vesicles and vesicles containing lysosomal enzymes from the Golgi apparatus. Endosomes can give rise to lysosomes, organelles that efficiently digest internalized material.

ER Endoplasmic reticulum.

Exocytosis A process by which cells secrete material via fusion of membrane-bound secretory vesicles with the plasma membrane.

Extracellular Matrix An elaborate system of proteins and polysaccharides that surrounds cells and tissues. Composed of structural elements, as well as soluble factors that influence cell growth, differentiation, and function.

FAK Focal adhesion kinase.

Flippase A membrane-associated enzyme that is able to transfer phospholipids between each layer of the lipid bilayer.

Focal Adhesion Kinase A protein kinase associated with focal contacts involved in transducing contact information at the cell surface into a cellular response.

G-Protein Small proteins involved in signaling functions that are able to bind GTP or GDP. G-proteins cycle between active and inactive states, depending on whether they are associated with GTP or GDP.

Gap Junction A type of cell-cell communicating junction that allows for the direct passage of small molecules between cells. Gap junctions are formed by the alignment of membrane-spanning pores, or connexons, between cells.

GDP Guanosine 5' diphosphate. A nucleoside formed by the hydrolytic removal of a phosphate group from GTP.

Golgi Apparatus The organelle associated with targeting protein and lipid synthesized by the ER to their appropriate locations. Material is transported between ER and Golgi, and between Golgi, lysosomes, and the plasma membrane by small membrane-bound vesicles. Modification of proteins and lipids (e.g., glycosylation) also occurs in the Golgi apparatus.

GTP Guanosine 5' triphosphate. A nucleoside associated with cell signaling and the regulation of cytoskeletal organization.

Hemidesmosome A type of adhesive junction that attaches epithelial cells to extracellular matrix. Bundles of intermediate filaments are connected to integral membrane proteins called integrins via linking proteins. Hemidesmosomal integrins bind to proteinaceous elements of the ECM, thereby providing mechanical linkage between IFs and the ECM.

IF Intermediate filament.

Inositol Triphosphate A type of membrane-associated phospholipid molecule involved in cell signaling.

Integral Membrane Protein A protein that passes through a lipid bilayer one or more times. Integral membrane proteins are strongly attached to membranes, and usually require disruption of membrane structure to be released.

Intermediate Filament One of the three major cytoskeletal groups of proteinaceous fibers found in cells. Associated with desmosomes and hemidesmosomes, they function in strengthening the cytoplasm of cells, as well as in mechanically linking cells of an epithelium with each other and with the ECM.

IP3 Inositol triphosphate.

LDL Low density lipoprotein.

Lipid Bilayer The structure phospholipid molecules adopt to form a membrane. Composed of two layers of phospholipid molecules, where the hydrophobic tails face each other and the polar heads face outward, creating a hydrophobic central region sandwiched between charged surfaces.

Lipid Raft A region of the plasma membrane exhibiting a specialized phospholipid makeup. Associated with specific functions, including formation of caveolae and signal transduction.

Lumen An enclosed space, or chamber. Usually refers to the compartment enclosed by a membranous organelle.

Lysosome A digestive organelle formed by the ER and Golgi apparatus. Hydrolytic enzymes synthesized by the ER are concentrated and packaged by the Golgi apparatus into lysosomal vesicles. Lysosomal vesicles fuse with either endosomes or with old cell organelles to digest internalized material, or cellular material to be recycled, respectively.

M6P Mannose-6-phosphate.

MRNA Messenger RNA; the type of RNA that encodes the sequence of amino acids to be assembled into a specific protein in association with a ribosome.

Mannose-6-Phosphate A polysaccharide "tag" attached to hydrolytic enzymes that marks them for packaging into lysosomal vesicles by the Golgi apparatus.

MAR Matrix attachment region.

Matrix Attachment Region A specialized sequence of DNA that is bound to the nuclear matrix.

Microfilament One of the three major cytoskeletal groups of proteinaceous fibers found in cells. Microfilaments are concentrated underneath the plasma membrane, which they support and help organize. They are also associated with adherens junctions, microvilli, and cleavage furrows. Microfilaments are involved with maintaining cell shape and powering cell motility.

Microvilli Finger-like extensions of the plasma membrane supported by core bundles of microfilaments. Microvilli serve to increase the absorptive area of epithelial cells.

Microtubule One of the three major cytoskeletal groups of proteinaceous fibers found in cells. Microtubules help organize the cytoplasm, participate in intracellular transport, allow for ciliary and flagellar motility, and organize and segregate chromosomes during mitosis.

Microtubule Organizing Center (MTOC) *see* **Centrosome**.

Mitochondria Double-membraned organelles primarily involved in converting the energy from food molecules into a form usable by the cell. This is largely accomplished by using food energy to create a proton gradient across the inner mitochondrial membrane, which in turn is used to drive the synthesis of ATP. Mitochondria also function in calcium homeostasis and in the regulation of programmed cell death, or apoptosis.

Mitosis The segregation of chromosomes during cell division. Cell division includes mitosis, followed by cytokinesis, or the division of the parental cell into two daughter cells.

MTOC Microtubule organizing center.

Nuclear Envelope A double-membraned structure enclosing the nucleus that establishes and maintains a distinct nuclear environment. The nuclear envelope is perforated by nuclear pores, which allow for the regulated transport of material between nucleus and cytoplasm.

Nuclear Matrix A protein (and possibly RNA) based scaffolding within the nucleus that is thought to help organize chromatin. The molecular makeup of the nuclear matrix is not well understood, but it appears to play important roles in DNA synthesis and the regulation of gene activity.

Nucleolus A specialized structure within the nucleus involved in ribosomal RNA synthesis and ribosome assembly.

Occludin An integral membrane protein associated with tight junctions that helps form a transcellular barrier between cells.

Occluding Junction Another term for tight junction (*see* **Tight Junction**).

Organelle Readily identifiable cellular inclusions that possess a characteristic morphology and function. The term is usually used to refer to the major membrane-bound structures within cells, including the nucleus, endoplasmic reticulum, Golgi apparatus, lysosomes, peroxisomes, and mitochondria.

Pericentriolar Material The amorphous material surrounding centrioles in a centrosome. Microtubules associated with centrosomes and spindle poles arise from the pericentriolar material.

Peripheral Membrane Protein A membrane-associated protein that is not embedded in the lipid bilayer. Peripheral proteins can be associated with the phospholipid heads of the bilayer, or with the portions of integral membrane proteins that extend beyond the bilayer.

Peroxisome Organelle involved in oxidation reactions, including the generation and destruction of H_2O_2.

Phagocytosis A type of endocytosis where very large particulate matter is taken up by a cell (e.g., the engulfment of bacteria by macrophages).

Pinocytosis A type of endocytosis where small vesicles internalize extracellular fluid for uptake by a cell.

Plasma Membrane The membrane surrounding a cell. The plasma membrane, sometimes called the plasmalemma, encloses the cytoplasm and protects the cell contents from the environment. It carries out vital functions in protection, transport of material into and out of the cell, sensing and responding to the environment, and in cell identification.

RER Rough endoplasmic reticulum.

Rough Endoplasmic Reticulum Endoplasmic reticulum that possesses ribosomes (thus presenting a “rough” surface). RER is primarily involved in protein synthesis.

Ribosome A multimeric array of protein and ribonucleic acid that is involved in protein synthesis. Ribosomes assemble individual amino acids into a polymer, or polypeptide, in a specific sequence determined by an associated messenger RNA molecule. Ribosomes can exist either “free” in the cytoplasm, or attached to the ER. In the former case, the proteins they produce are released into the cytoplasm; in the latter case, the proteins are either inserted into the membrane of the ER, or released into the lumen of the ER.

RNA Ribonucleic acid. Includes a number of subtypes, including messenger RNA (mRNA), ribosomal RNA (rRNA), and transfer RNA (tRNA).

SER Smooth endoplasmic reticulum.

Signal Recognition Particle An assembly of proteins that help dock ribosomes to ER (forming RER).

Smooth Endoplasmic Reticulum Endoplasmic reticulum that lacks ribosomes (thus presenting a “smooth” surface). SER is involved in lipid synthesis, calcium transport, and detoxification reactions.

SNAREs Proteins that help regulate the trafficking of membrane bound vesicles between different organelles.

S-Phase The stage of the cell cycle where DNA synthesis occurs.

Spindle Pole A microtubule organizing center that assembles the microtubule spindle during cell division. When cells divide, their centrosome duplicates and moves apart to form a bipolar spindle.

Spliceosome An assembly of protein and RNA molecules that processes newly-made mRNA into mature mRNA.

SRP Signal recognition particle.

Start-Transfer Sequence A specific sequence of amino acids that initiates the penetration of a growing polypeptide chain into the lipid bilayer of RER.

Stop-Transfer Sequence A specific sequence of amino acids that stops the insertion of a polypeptide chain into the lipid bilayer of RER.

Synapse A type of communicating cell-cell junction found between neurons in nervous tissue.

Tight Junction A type of cell-cell junction that establishes a transcellular barrier. Also referred to as an occluding junction. The barrier is established by the integral membrane proteins occludin and claudin, which are arranged in strands that adhere to each other, and form an apical network around epithelial cells.

***trans*-Golgi** The portion of the Golgi apparatus that releases membrane-bound vesicles after their contents have been processed by the Golgi.

Tubulin The protein used to form microtubules. Most microtubules continually oscillate between growth and disassembly by the addition or removal of soluble tubulin dimers at the tubule termini.

WASp The protein mutated in Wiscott-Aldrich syndrome patients. This protein plays important roles in regulating actin polymerization.

Zonula Adherens A large, belt-like adherens junction usually found encircling the apical regions of epithelial cells.

Suggested Reading

Plasma Membrane

Fielding, C. J. and Fielding, P. E. (2000) Cholesterol and caveolae: structural and functional relationships, Biochim. Biophys. Acta. 1529, 210–222.

Fleming, T. P., Ghassemifar, M. R., and Sheth, B. (2000) Junctional complexes in the early mammalian embryo, Semin. Reprod. Med. 18, 185–193.

Gagescu, R., Gruenberg, J., and Smythe, E. (2000) Membrane dynamics in endocytosis: structure—function relationship, Traffic 1, 84–88.

Ikonen, E. (2001) Roles of lipid rafts in membrane transport, Curr. Opin. Cell. Biol. 13, 470–477.

Lee, A. G. (2000) Membrane lipids: it's only a phase, Curr. Biol. 10, R377–380.

Mukherjee, S. and Maxfield, F. R. (2000) Role of membrane organization and membrane domains in endocytic lipid trafficking, Traffic 1, 203–211.

Simons, K. and Toomre, D. (2000) Lipid rafts and signal transduction, Nat. Rev. Mol. Cell. Biol. 1, 31–39.

Telo, P., Lostaglio, S., and Dejana, E. (1997) Structure of intercellular junctions in the endothelium, Therapie. 52, 395–398

von Heijne, G. (1999) Recent advances in the understanding of membrane protein assembly and structure, Q. Rev. Biophys. 32, 285–307.

Endoplasmic Reticulum; Golgi Apparatus; Lysosomes

Agarraberes, F. A. and Dice, J. F. (2001) Protein translocation across membranes, Biochim. Biophys. Acta. 1513, 1–24.

Baumann, O. and Walz, B. (2001) Endoplasmic reticulum of animal cells and its organization into structural and functional domains, Int. Rev. Cytol. 205, 149–214.

East, J. M. (2000) Sarco(endo)plasmic reticulum calcium pumps: recent advances in our understanding of structure/function and biology, Mol. Membr. Biol. 17, 189–200.

Ellgaard, L. and Helenius, A. (2001) ER quality control: towards an understanding at the molecular level, Curr. Opin. Cell. Biol. 13, 431–437.

Glick, B. S. (2000) Organization of the Golgi apparatus, Curr. Opin. Cell. Biol. 12, 450–456.

Griffiths, G. (2000) Gut thoughts on the Golgi complex, Traffic. 1, 738–1745.

Lafontaine, D. L. and Tollervey, D. (2001) The function and synthesis of ribosomes, Nat. Rev. Mol. Cell. Biol. 2, 514–2520.

Lippincott-Schwartz, J. and Zaal, K. J. (2000) Cell cycle maintenance and biogenesis of the Golgi complex, Histochem. Cell. Biol. 114, 93–103.

Pshezhetsky, A. V. and Ashmarina, M. (2001) Lysosomal multienzyme complex: biochemistry, genetics, and molecular pathophysiology, Prog. Nucleic Acid Res. Mol. Biol. 69, 81–114.

Yew, N. S. and Cheng, S. H. (2001) Gene therapy for lysosomal storage disorders, Curr. Opin. Mol. Ther. 3, 399–406.

Mitochondria

Frey, T. G. and Mannella, C. A. (2000) The internal structure of mitochondria, Trends Biochem. Sci. 25, 319–324.

Gottlieb, R. A. (2001) Mitochondria and apoptosis, Biol. Signals Recept. 10, 147–161.

Martinou, J. C. and Green, D. R. (2001) Breaking the mitochondrial barrier, Nat. Rev. Mol. Cell. Biol. 2, 63–67.

Rassow, J. and Pfanner, N. (2000) The protein import machinery of the mitochondrial membranes, Traffic 1, 457–464.

Szibor, M., Richter, C., and Ghafourifar, P. (2001) Redox control of mitochondrial functions, Antioxid. Redox. Signal. 3, 515–523.

Peroxisomes

Hayashi, S., Fujiwara, S., and Noguchi, T. (2000) Evolution of urate-degrading enzymes in animal peroxisomes, Cell. Biochem. Biophys. 32, 123–129.

Wanders, R. J., Vreken, P., Ferdinandusse, S., Jansen, G. A., Waterham, H. R., van Roermund, C. W., and Van Grunsven, E. G. (2001) Peroxisomal fatty acid alpha- and beta-oxidation in humans: enzymology, peroxisomal metabolite transporters and peroxisomal diseases, Biochem. Soc. Trans. 29, 250–267.

Cytoskeleton

Banwell, B. L. (2001) Intermediate filament-related myopathies, Pediatr. Neurol. 24, 257–263.

Feng, Y. and Walsh, C. A. (2001) Protein-protein interactions, cytoskeletal regulation and neuronal migration, Nat. Rev. Neurosci. 2, 408–416.

Morgan, K. G. (2001) Nonmuscle motility/cytoskeleton, Am. J. Physiol. Cell. Physiol. 280, C1634–C1635.

Nogales, E. (2001) Structural insight into microtubule function, Annu. Rev. Biophys. Biomol. Struct. 30, 397–420.

Powledge, T. M. (2001) Tensegrity lives, J. Cell. Biol. 154(2), 254.

Reilein, A. R., Rogers, S. L., Tuma, M. C., and Gelfand, V. I. (2001) Regulation of molecular motor proteins, Int. Rev. Cytol. 204, 179–238.

Scholey, J. M., Rogers, G. C., and Sharp, D. J. (2001) Mitosis, microtubules, and the matrix, J. Cell. Biol. 154, 261–266.

Stidwill, R. P. and Greber, U. F. (2000) Intracellular virus trafficking reveals physiological characteristics of the cytoskeleton, News. Physiol. Sci. 15, 67–71.

Nucleus

DePamphilis, M. L. (2000) Review: nuclear structure and DNA replication, J. Struct. Biol. 129, 186–197.

Houtsmuller, A. B. and Vermeulen, W. (2001) Macromolecular dynamics in living cell nuclei revealed by fluorescence redistribution after photobleaching, Histochem. Cell. Biol. 115, 13–21.

Imamoto, N. (2000) Diversity in nucleocytoplasmic transport pathways, Cell. Struct. Funct. 25, 207–216.

Jackson, D. A. (2000) Features of nuclear architecture that influence gene expression in higher eukaryotes: confronting the enigma of epigenetics, J. Cell. Biochem. 35, 69–77.

Kiseleva, E., Goldberg, M. W., Cronshaw, J., and Allen, T. D. (2000) The nuclear pore complex: structure, function, and dynamics, Crit. Rev. Eukaryot. Gene. Expr. 10, 101–112.

Sotolongo, B. and Ward, W. S. (2000) DNA loop domain organization: the three-dimensional genomic code, J. Cell. Biochem. 35, 23–26.

Textbooks

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., and Watson, J. D. (1994) Molecular Biology of the Cell, 3rd. ed., Garland Publishing, Inc., New York, NY.

Cooper, G. M. (2000) The Cell, A Molecular Approach, 2nd. ed., Sinauer Associates, Inc., Sunderland, MA.

3 Cell Signaling

Daniel A. Rappolee

Introduction

Intercellular communication is essential for development and homeostatic function in multicellular organisms. The language of intercellular communication takes many different forms. These include protein growth factors e.g., local paracrine, like Fibroblast growth factor (FGF), blood-borne endocrine growth hormone; hydrophobic steroids, e.g., estrogen; lipid mediators, prostaglandins, leukotrienes; modified amino acids, e.g., neurotransmitters such as adrenaline and other metabolites, e.g., nitric oxide. This chapter will focus largely on small protein growth factors that modulate growth, differentiation, apoptosis and steady state function during development and in the adult.

An essential role of signal-transduction is to coordinate functions of diverse cell types and sets of identical cells within an organ that require close and synchronous activity in the multicellular organism. The speed of intercellular communication is dependent on distance and the mode of delivery of the intercellular signal. Local or paracrine intercellular communication acts within milliseconds over distances less than 10–20 cell diameters (approx 200 microns), but endocrine or blood-borne signaling that occurs over a distance of meters requires minutes. Specialized short distance signaling, like that mediated by gap junctions, allows linked cells to share small intracellular signal-transduction intermediates downstream of receiving cell-surface receptors. Specialized long distance signaling, over a distance of meters, can be expedited to the millisecond range, by increasing the conductance speed. This is accomplished by saltatory movement of signals in neurons coupled with the fast action of neurotransmitters at post-synaptic membranes.

Once the interactor, i.e., the ligand, has bound to its receptor, intracellular signal-transduction is initiated. After the receptor is activated by phosphorylation, receptor binding of nonenzymatic docking molecules like IRS-1 (insulin-receptor substrate) peaks within 1 min. At the end of this signal transcription pathway, phosphorylation of transcription AP-1 factor initiates transcription of c-fos within 1 min which peaks within 15 min. Other hormones, such as estrogen activate transcription over slower time periods, approaching 1 h. In some signal-transduction processes, such as vision, the complete process of photons activating rhodopsin is computed in one second.

The Receptor

If the ligands are the words of intercellular language, the receptors are the *ears*. Trans-membrane plasmalemma receptors are of several types:

1. Receptor tyrosine kinases such as the FGF receptor;
2. Serine-threonine kinase receptors such as the TGF-β receptor;
3. Nonenzymatic transmembrane receptors (e.g., integrins) that are linked to intracellular tyrosine kinases (e.g., focal adhesion kinase [FAK] and proline rich tyrosine kinase tyrosine kinases [PYK2]);
4. Seven transmembrane spanners receptors linked to G-proteins (e.g., Wnt ligand-wingless and int oncogene- and its receptor; frizzled);
5. Non-enzymatic receptors that are linked to signaling pathways which are derepressed by allosteric-conformational changes (e.g., smoothened receptor for hedgehog ligand);
6. Receptors which are proteolytically converted to ligand (e.g., the Notch intracellular C terminal domain).

Another class of signaling receptors is located in the cytoplasm. They bind hydrophobic hormones such as steroids (e.g., estrogen), thyroid hormones, and retinoic acid ligands. They are then translocated to the nucleus; where they act as transcription factors.

The Signal Cascade

Signaling cascades within the cell start as allosteric changes in the receptor or nonenzymatic docking proteins (e.g., the IRS-1 family and FRS2). These convey the signal by conformational change and by becoming targets for phosphorylation by receptor kinases. In either case, the function of the signal-transduction pathway is to quickly amplify, and directionally conduct information reaching the cell by transmitting it through a series of tyrosine and serine-threonine kinases.

The majority of the transmembrane receptors are in the *off* state until induced by an extracellular ligand. At this point receptors act like allosteric enzymes with the enzyme in the cytoplasmic domain. Many of the tyrosine kinases require multimer formation as there is not sufficient flexibility in the transmembrane alpha-helix to mechanically transduce the ligand-induced conformational change. Multimerization brings together cytoplasmic enzyme domains that cross-activate and then signal downstream docking and enzymatic signal-transduction proteins.

The receptor is activated for a period of time before it is destroyed (internalized and degraded), desensitized (by phosphorylation by a receptor induced kinase such as β-adrenergic receptor kinase), or dephosphorylated. During the activation period, a single activation event can lead to a highly amplified signal-transduction event. For example, a single quantum of light activating the photoreceptor rhodopsin leads to the hydrolysis of one hundred thousand cGMP signaling molecules for the duration of 1 s. The activated receptor has multiple possible phosphorylation sites on the cytoplasmic domains. These are capable of interacting with large numbers of signaling intermediates that *see* the activated receptor through the *src* homology domain-2 (SH2) that bind phosphorylated tyrosines (*see* Fig. 1).

This branching of pathway choices immediately downstream of the FGF receptor is summarized in Figs. 1 and 2. The array of these docking proteins distinguishes different cell types that may each express the same receptor and facilitates different

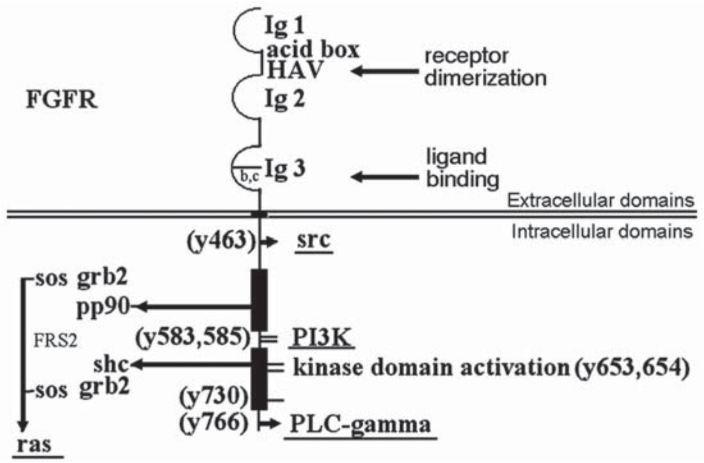


Fig. 1. The FGF receptor. The FGF receptor is an allosteric enzyme with the allosteric sites in the ectodomain and the enzymatic tyrosine kinase in the cytoplasm. Many mapped potentially functional and functionally active tyrosine activation sites are known.

signal types in cells with the same receptor. If the activated receptor is the hub of activity, the docking proteins provide the spokes that radiate out to multiple downstream targets. Docking proteins such as IRS-1 and FRS-2 (insulin and FGF receptor substrate) are receptor-binding proteins that can initiate many branches of the signal-transduction pathway. The importance of the additive effect of branched pathways is indicated by recent studies analyzing the effects of mutating the phosphorylation/docking sites of the PDGF receptor on the activation of sets of newly transcribed genes (Hill and Treisman, 1999). Sets of newly transcribed genes have been analyzed for quality and magnitude of the induction using cDNA-based microarrays. The results suggest that more than one phosphorylation/docking site on the PDGF receptor is needed for the full and proper magnitude and breadth of the transcriptional response.

There are three MAPK families (*see* Figs. 2 and 3, Table 1), each having its own nonenzymatic signal-transduction intermediate before the G protein/*ras* signal-transduction intermediate that is itself a GTPase. Thus, *ras* is a key component in mitogenic signal-transduction. There are 4 sequential vertical tiers of serine-threonine kinases (*see* Table 1); each can have different *horizontal* interacting components. In the three MAPK signal-transduction pathways, the first two serine-threonine kinases have only cytoplasmic targets, whereas the last two tiers have both cytoplasmic targets and nuclear targets. Therefore, the activation of these last two tiers of serine-threonine kinases (MAPKs and MAPKAPS) can lead to nuclear localization, transcription factor phosphorylation (e.g., Elk1 and ATF2 for ERK1,2) and gene transcription. Other nontyrosine kinase initiated pathways also have homologous cytoplasmic signal transducers. For example, although it is thought that Raf-1 mediates the effects of the FGF receptor (*see* Fig. 2, Table 1), Raf-1 homologs in the TGF-β pathway (TAK-1; Fig. 3, Table 1), and seven-spanner G-protein pathway (RafB; Fig. 3, Table 1) also act as cytoplasmic signal transducers. Branching of the pathway can occur at any tier of the vertical cascade (*see* Table 1). Regulation of branching also distinguishes cell types and experience of the cell.

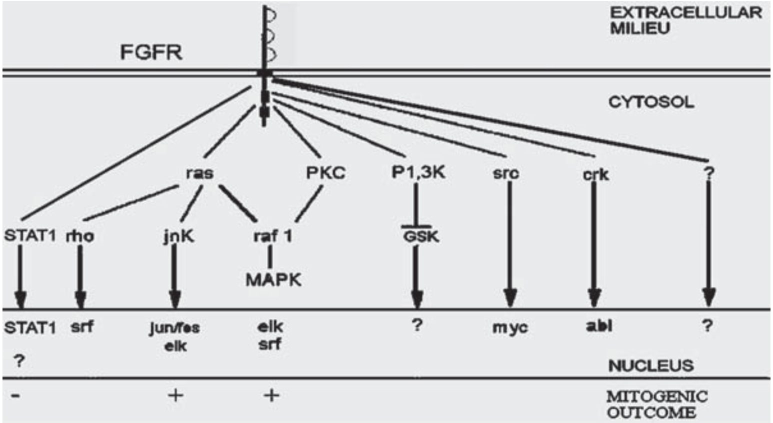
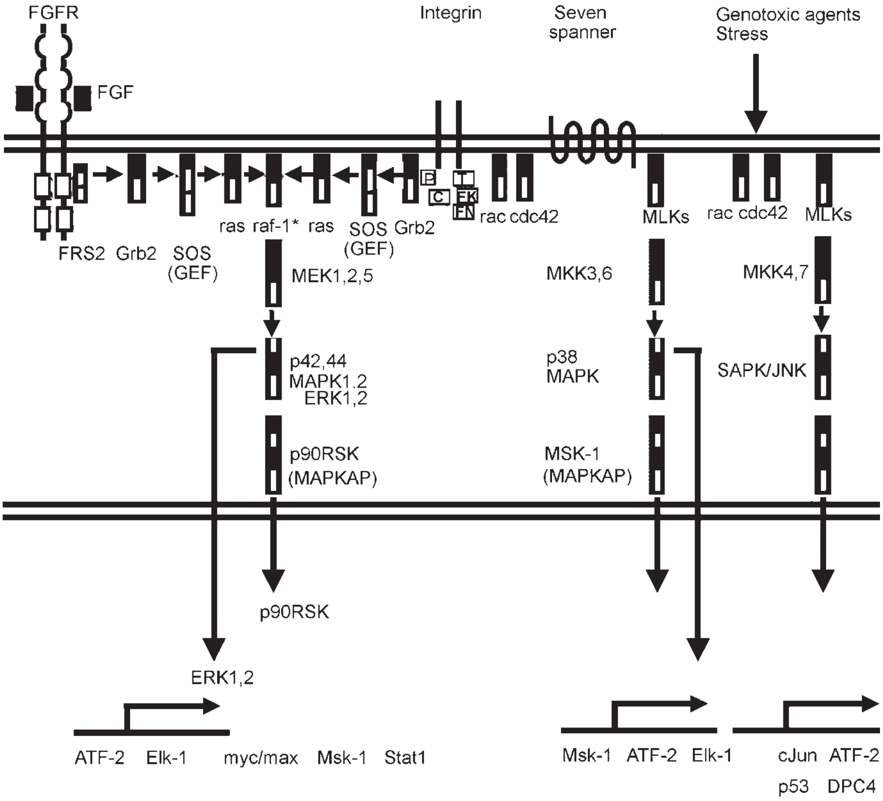


Fig. 2. Mitogenic and transcription activating signal-transduction pathways downstream of FGFR. A preponderance of evidence in cell lines suggests that the *ras*-MAP kinase pathway mediates the mitogenic signal of FGFR. Note that the *src*, crk are only indirectly implicated in FGFR signal-transduction due to sequence homology with other receptors and possible binding sites in the cytosolic domain. Jun kinase and P1,3 kinase can be mitogenic in certain circumstances, but mutation of the P1,3 kinase activating site in an FGFR in vitro did not prevent a mitogenic response to FGF. The STAT1 pathway has recently been shown to be anti-mitogenic in FGFR-3 mediated chondrocyte cell division cessation. The + indicate the most likely pathways for FGFR cell-division control.



Turning Off the Response

Signaling requires expeditious and tight control to maintain homeostasis and to ensure proper development. Control is exerted at all tiers of the pathway and at various levels of production and activation of the signal-transduction proteins. The receptor is activated for a period of time before it is destroyed, desensitized, or dephosphorylated. Other tiers of the pathway are regulated in similar ways. Ras is inactivated by GTPase activating proteins (GAPs), MEK family members are dephosphorylated and inactivated by protein phosphatase (PP)-1 and PP2A, ERK family members are dephosphorylated and inactivated by MAP kinase phosphatases MKP-3 and MKP6, and JNK is inactivated by M3/6. The mRNA transcripts for many of the signal-transduction genes have a consensus destruction sequence in the 3'- untranslated region that confers a short half-life. Rapid regulation is achieved at the levels of protein and mRNA stability, protein activation, and signal-transduction.

Signal Transduction Pathways

A list of signal-transduction pathway websites is presented in Table 2. Information can be obtained on the activation of PKA and PKC, the mechanism of Calcium-calmodulin signaling, prostaglandins and leukotrienes, and nitric oxide. Other cytostatic pathways mediated by JAK-STAT receptors for the interferon-γ (IFN-γ) receptor, and apoptosis pathways through tumor necrosis factor-α (TNF-α) are also included. Noncanonical pathways mediated by serine-threonine receptor kinases (TGF-β receptor), and novel pathways for signaling by derepressing seven spanner receptors (Hedgehog-ligand derepression of patched receptor by smoothened) are also included. A novel signaling mechanism important in development, signaling by proteolytically cleaved ligand-activated receptor (Delta ligand activation of the Notch protein that cleaves and translocates the protein to the nucleus), is at the Science website for Signal Transduction Knowledge environment (*see* Website: http://www.stke.sciencemag.org). The Wnt-frizzled-GSK3-β-catenin pathways interaction with E-Cadherin-β-catenin are not discussed in this chapter. FGF receptor, integrin, and G protein- activating receptors and their activation of the three MAPK families (*see* Figs. 2 and 3) can be used to illustrate the basic principles of signal-transduction pathways.

Fig. 3. *(opposite page)* The MAPK family. Each member is embedded in enzymatic cascades of intracellular serine-threonine kinases regulated by tyrosine kinases and allosteric docking proteins. The pathways initiate at the plasmalemma by receptor tyrosine kinases, ECM-binding integrins, and G-protein binding receptors. ATF-2- Activating transcription factor 2, CRE-BP1, CREB2; Cas Crk-associated substrate, p130CAS (**C in integrin signaling complex); c-Raf Raf proto-oncogene S/T protein kinase; DPC-4- Deleted in pancreatic cancer locus 4, SMAD4; ELK1-Ets domain transcription factor, ERK-Extracellular signal-regulated kinase, MAPK; FAK- Focal adhesion kinase (**FK in integrin signaling complex); FGF-fibroblast growth factor; FGFR-fibroblast growth factor receptor; FRS2- FGF receptor stimulated, lipid-anchored Grb2 binding protein; Fyn- src family tyrosine kinase (**Fn in integrin signaling complex); GEF Guanine nucleotide exchange factor (example is SOS son-of-sevenless); GRB2-Growth factor receptor-bound protein 2; JNK-Jun N-terminal kinase; Jun-transcription factor; MAPK-Mitogen-activated protein kinase; MAPKAP- MAP kinase-activated protein kinase 2; MEK-MAPK/ERK kinase, MAPKK; MKK MEK kinase; MLK-Mixed lineage kinase; MSK-1- Mitogen and stress-activated kinase 1; p53 Tumour suppressor protein that protects from DNA damage; Paxilin (P in integrin signaling complex); PYK2-Proline-rich tyrosine kinase-2 (**P in integrin signaling complex); rac-G-protein; ras-G-protein; RSK- Ribosomal S6 kinase; SAPK- Stress-activated protein kinase; STAT- Signal transducer and activator of transcription; Talin (**T in integrin signaling complex).

Table 1
Extracellular to Intracellular Signal-Transduction for Three MAPK Families: Levels of Signal-Transduction in the Cascades of the Three MAPK Families[a]

Ligand	FGF1-23	Other ligands		
↓				
Receptor	FGFR1-4	Other receptors		
↓				
Nonenzymatic Docking molecule 1 RTK binding	FRS2	GAB1,2	IRS1-4	
↓				
Nonenzymatic Docking molecule 2 RTK binding Docking molecule 1 binding	Grb2	shc	shb	
↓				
Ras superfamily Regulatory molecules	SOS (GEF family)	GRF family Rate limiting	GAP family	vav
↓				
Ras superfamily G protein family Enzyme	*ras*	Rac Rho	dc42 Rap1	
↓				
Kinase 1 (MEKK) Enzyme	Raf-1	RafB KSR	Tak1	
↓				
Kinase 2	MEK1, 2, 5	MEK3,6	MEK4,7	
↓				
Enzyme	MAPK family One	MAPK family Two		MAPK family Three
↓				
Kinase 3	p42, 44 ERK1,2 MAPK1,2, MAPK5	p38MAPK	SAPK/JUNK enzyme	
↓				
Kinase 4 Enzyme	p90RSK	MAPKAP	MNK1,2	MSK1
↓				
Resident or Translocated nuclear Factors	Elk1 myc/max pRSK90 MAPK family ATF-2 STAT1 Jun			

[a] FGF signaling is used as an example. *See* Fig. 3 caption for abbreviations.

Table 2
Signal Transduction Websites:
Electronic Resources for Signal-Transduction Reagents and Information

Website: http://stke.sciencemag.org/
Signal-transduction knowledge environment (STKE). Excellent resource for broad and focused signal-transduction electronic and archival published literature. PDFs and full text articles with JPG figures are available. Requires AAAS membership and an STKE users fee.

Website: http://kinase.oci.utoronto.ca/signallingmap.html
Very good focus with map of the 3 MAPK families and clickable short to long descriptions of molecules on the map.

Website: http://www-personal.umich.edu/%7Eino/List/
Good outline of signal-transduction pathways with links to PubMed discovery articles.

Website: http://www.grt.kyushu-u.ac.jp/spad/index.html
Good, clickable diagrams, but not recently updated.

Website: http://vlib.org/Science/Cell_Biology/signal_transduction.shtml
Good cross-referenced site for information about function and sequence references for signal-transduction genes.

Website: http://www.cellsignal.com/
Company site. Short description of signaling intermediates in the literature and available antibodies. Also, check sections under pathway diagrams for other web resources .

Website: http://www.clontech.com/gfp/
Company site for signal-transduction expression transgenes.

Website: http://www.scbt.com/
Company site. Short description of signaling intermediates in the literature and available antibodies. Also, check sections under pathway diagrams for other web resources.

Receptor Tyrosine Kinases (FGF Receptor) and Mitogenesis

Ligand-dependent autophosphorylation and activation of the FGFRs signals into four pathways leading to new transcription in cell lines (*see* Figs. 1 and 2). Two of these, p38MAPK and Jun kinase, are generally not mitogenic in cell lines. The major FGFR mitogenic signaling pathway is the *ras*-Raf-1-MAPK pathway, also known as the *universal cassette* because of the weight of evidence for its mitogenic role in diverse cell lines. FGF activation of MAPK (ERK-1, ERK-2) is necessary and sufficient to activate transcription factors elk and SRF, leading to new transcription and a strong mitogenic response in cell lines.

A second pathway that mediates mitogenesis through the FGFRs leads to binding of phospholipase C (PLC)-γ and Ca^{2+}-dependent PI-3-kinase through phosphorylation of other unique tyrosine residues and the subsequent activation of phosphoinositol turnover, generation of diacylglycerol and the activation of PKC. There are three groups of PKC; conventional (α,β,γ), novel (η, ν, θ), and ξ) atypical (λ). The ξ and atypical families are not mitogenic and are brain-specific. Activation of PKC-α,β, and γ leads to an increased mitogenic response, primarily through Raf-1 and MAPK, although this appears to be less important than the *ras*-dependent MAPK pathway. Substitution of the tyrosine on FGFR-1 responsible for PLC-γ binding and activation of PKC does not diminish FGF-dependent mitogenesis, suggesting that PKC is not necessary for mitogenesis. However, in studies with a related PDGF receptor, mutations causing inactivity in all tyrosine sites for mitogenic effects were rescued by the inclusion of the the tyrosine that activates the PLC-PKC pathway. This suggested that the PKC pathway can be sufficient for mitogenesis. It is important to note that PKC and *ras* activate Raf-1 by a separate mechanism and that Raf-1 activates mitogenesis via MAPK activity. Therefore, the FGF receptor can activate Raf-1 through either pathway, although *ras* is most powerful. Recent analysis of the *raf-1* null mutant has suggested that *ras* is the most important, but that the Raf-B, not Raf-1 is necessary for growth factor mediated mitogenesis.

As observed in cell lines, activation of *src* is a third possible pathway of FGFR cell cycle activation. The FGFR activation of *src* (*see* Fig. 1) has not been found to be mitogenic for FGF, but is mitogenic in other cell lines and mediates functions like cell scattering and activation of nuclear transcription during PC-12 differentiation.

A fourth pathway where FGF activity suppresses cell division through STAT1 has been identified. In the gain-of-function mutation leading to sporadic nonfamilial dwarfism, a single change in a transmembrane amino acid in FGFR-3 results in a gain-of-function enzymatic activity that leads to cessation of the cell cycle in chon-drocytes. Recently, suppression of chondrocyte division was shown to require STAT1 activity, but the mode of activation of STAT1 is not understood. The expression of the STAT1 pathway has not been tested in preimplantation embryos. However, STAT1 should be considered when interpreting results after perturbing FGF receptors.

MAPK Families

The structure of the cascading biochemical pathway of the three MAPK family pathways docking proteins and kinases, are very similar (*see* Table 1, Fig. 3). However, functionally, the MAPK/ERK pathways are more mitogenic and the p38MAPK and SAPK/JNK pathways are cytostatic. The mechanistic basis of the separation of function is not clear, but is based more on the quantity of each type of transcription factor activated than the quality. Each MAPK family activates a large overlapping group of transcription factors (*see* Fig. 3), but overexpression of receptors (increasing pathway strength), can change the outcome of mitogenesis to one of differentiation within a single cell type. Early studies examined functionally similar receptors (mitogenic FGF and PDGF) and did not compare receptors that mediate more diverse biological outcomes. In comparison, recent studies using cDNA microarrays have suggested that different factors result in transcriptional activation of similar sets of genes. Differences in strength of transcription is an important difference. The use of cDNA microarrays to analyze intermediate transcriptional outcomes (primary and secondary waves of induced transcription) with respect to upstream receptor signaling capacity and downstream biological outcome, will yield significant insights into the function of signaling pathways.

As described in the section “The Signal Cascade”, the three MAPK family pathways go through similar tiers of nonenzymatic docking proteins and serine threonine kinases. Each of these tiers branch to affect cytoplasmic targets, but only the last two tiers of proteins migrate to the nucleus and directly affect transcription factors (*see* Table 1; Fig. 3).

A long initial phase of research in signal-transduction has focused upon identifying novel signaling intermediates and their roles in the various pathways. The three MAPK families illustrate common features of related and interacting signal-transduction pathways: speed of transduction, amplification via cascading enzymatic activity, branching and interaction with other pathways, and distinct and shared biologic function mediated through shared and distinct and shared transcription factors. The next phases of research will focus on the function of all members within a family of signal-transduction genes as the human and mouse genome projects provide complete sets of family members. Large-scale approaches paint a broader picture of the responses of cells to ligands that induce distinct biological outcomes, such as cell death, mitosis, or motility. In this case, microarrays will be utilized to detect broad changes in signaling quantity and quality of the sets of transcription factors between normal cells and null mutants or cells with receptor mutants with differential signal-transduction capacities. For example, differences in transcriptional quantity and quality of functionally different receptors that activate mitogenesis through FGF receptor-ERK signaling, that block mitogenesis through TGF-β receptor-SMAD (contraction of Sma and Mad [Mothers against decapentaplegic] genes) signaling or induce apoptosis through TNF-α receptor- Fas-associated protein with death domain (FADD) signaling, are sure to be studied.

Glossary and Abbreviations

ATF A family of activating transcription factors (ATF) binding both AP-1 and CRE response elements in promoters and using leucine zipper protein interaction domains. These transcription factors are targets of stress, UV, and viral infection (*see* Fig. 2 and Suggested Reading).

ERK (MAPK family) One of three families of serine threonine kinases designated extracellular-signal regulated kinases (ERK) that mediate the largely mitogenic signals of extracellular growth factors/hormones and neurotransmitters. There are currently five members of the family; ERK1, ERK2 (aka MAPK1 or p42, and MAPK2 or p44), ERK3, ERK5, and ERK 6. Phosphorylation of threonine and tyrosine residues by upstream MEK1, 2, and 5 is required for activation. Activation by phosphorylation is followed by nuclear translocation and phosphorylation of transcription factors such as Elk-1 (*see* Fig. 2 and Suggested Reading).

Grbs SHc SHb Docking proteins with SH2-phospho-tyrosine binding domains that directly bind activated receptors or receptor activated docking proteins like IRS and FRS family docking intermediates. These proteins then interact with GRFs, GEFs, and GAPs to modulate downstream *ras* family activity. Grb2 is essential for survival of very early mammalian development.

GTPase releasing factors (GRF), SOS enhancing factors (GEF), and activating factors (GAP) A large family of molecules that enhance the activity of *ras* by facilitating its existence in the GTP bound state. These activating factors act by ridding inactive *ras* of bound GDP (GDP-release factors or GRFs), and by promoting binding of GTP to *ras* to activate it (GTP exchange factors or GEFs). The inactivating factors activate *ras* inherent GTPase activity, converting *ras*-GTP into inactive *ras*-GDP (GTPase activating protein- GAPs).

IRS1-4, FRS-2 Nonenzymatic, docking molecule families that directly bind the phosphorylated and activated insulin receptor and FGF receptors. Upon binding of tyrosine phosphorylated receptors through SH2 or PBZ domains, each receptor substrate has multiple tyrosines that can be phosphorlated and provide docking sites for downstream SH2 or PBZ domain expressing signal-transduction proteins.

MAPKAP MAPK Activated Protein Kinases that are activated by the three MAPKs subsequently have cytoplasmic targets or migrate to the nucleus. The family includes MSK-1, Mnk, and RSK family members.

MEK A family of homologous mixed-function kinases downstream of the Raf1 family of serine threonine kinases and upstream of the three MAP-ERK kinase families. There are currently seven members of the family. The prototype, MEK1, specifically phosphorylates MAPK/ERK1,2 on both regulatory threonine and tyrosine residues. MEK 2-7 phosphorylate and activate all three MAPK family serine threonine kinases in the following interactions; MEK2:ERK1,2, MEK3:p38MAPK (aka MKK3), MEK4:p38MAP or MEK:JNKMAPK (aka MKK-4), MEK5:ERK5, MEK6:p38MAPK, and MEK7:JNKMAPK (*see* Fig. 2 and Suggested Reading).

MSK-1 Mitogen activated S6 kinases (MSK) are MAPKAPs (MAPK-activated protein kinase) that is activated by ERK as well as p38 MAPK in response to growth factors and cellular stress. There are currently three members of the family. MSK protein is similar to MAPKAP RSK in having two kinase domains on either side of a regulatory domain (*see* Figs. 2 and 3 and Suggested Reading).

p38 (MAPK family) One of three families of related MAPKs designated p38 MAPK. p38 is more similar to SAPK/JNK in its activation by stress, UV, and cytokines/inflammation more than growth factors. Related proteins include SAPK4 and yeast HOG1. Targets include max transcription factor, ATF-2 and MAPKAP-2.

Raf family (aka MAPKKK) A family of serine threonine kinases that are activated on separate domains downstream to *ras* and PKC (protein kinase C). There are three family members Raf-1, RafA, and RafB and two related proteins (below). Raf-1 is stimulated downstream to receptor tyrosine kinases and *src*. Raf activates MEKs and is required for activation of the ERK/MAPK pathway. TAK-1 (TGF-β activated kinase) and KSR (kinase suppressor of *ras*) are related serine threonine kinases.

Ras-G proteins (*ras*-Rac-Rho-Rap1,2-Ral) GTPase enzymes that activate Raf and other downstream cytoplasmic signaling kinases while in their GTP bound state. Cellular localization is essential to function and *ras* family proteins are localized to plasmalemma, cytoskeleton, and vesicular membranes. Currently includes three *ras* family members, three Rap1 and Rap2 family members, two Ral family members; six Rac, Rho, and cdc42 family members, and three Rho family members.

RSK Ribosomal S6 kinase (RSK) are MAPKAPs (MAPK activated protein kinase) that are activated by mitogens and stress. There are currently three family members Rsk-1 (aka p90RSK), Rsk-2, and Rsk-3. RSKs were initially thought to have ribosomal S6 protein as a target, but now nuclear transcription factors and proteins such as histone H3 are thought to be the physiologic targets (*see* Figs. 2 and 3, Suggested Reading).

SAPK/JNK (MAPK family) One of three families of related MAPKs designated stress activated protein kinase (SAPK)/cJun kinase (JNK). They are related to ERK/MAPK in requiring activation by phosphorylation of nearby tyrosine and threonine residues by upstream MEKs. They are activated by stresses such as UV and viral infection, not mitogens. There are currently three family members; JNK-1-3. (*see* Fig. 2 and Suggested Reading).

STAT STATs are a family of 6 currently known transcription factors that bind growth factor receptor tyrosine kinases (e.g., STAT1 binds directly FGF receptor) and cytokine activated IFN-γ receptors that activate STATs through the JAK intracellular non-receptor tyrosine kinases.

Suggested Reading

Introduction; Background and Reviews

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J. D. (eds.) (1994) Molecular Biology of the Cell, 3rd ed. Garland Publishing, NY, p. 755.

Angel, P. and Karin, M. (1991) The role of Jun, Fos and the AP-1 complex in cell-proliferation and transformation, Biochem. Biophys. Acta. 1072, 129–157.

Cherfils, J. and Chardon, P. (1999) GEFs: Structural basis for their activation of GTP-binding proteins, Trends Biochem. Sci. 24, 306–311.

Heim, M. H. (1999) The Jak-STAT pathway: cytokine signalling from the receptor to the nucleus, J. Recept. Signal. Transduct. Res. 19, 75–120.

Murakami, M. and Morrison, D. K. (2001) Raf-1 Without MEK? Science STKE, 1–3.

Myers, M. G., Wang, L.-M., Sun, X. J., Zhang, Y., Yenush, L., Schlessinger, J., et al. (1994) Role of IRS-1-GRB-2 complexes in insulin signaling, Mol. Cell. Biol. 14, 3577–3587.

Nishida, E. and Gotoh, Y. (1993) The MAP kinase cascade is essential for diverse signal transduction pathways, Trends Biochem. Sci. 18, 128–131.

Perrimon, N. and Perkins, L. (1997) There must be 50 ways to rule the signal: The case of the Drosophila EGF receptor, Cell 89, 13–16.

Rappolee, D. A. and Werb, Z. (1994) The Role of Growth Factors in Early Mammalian development, Advances in Developmental Biology 3, 41–71.

Rappolee, D. A. (1998) Growth factors and hormones in mammalian development. Growth factors in the mammalian pre- and post-implantation embryo, in: Hormones and Growth Factors in Development and Neoplasia, (Dickson, R. and Salomon, D., eds.) John Wiley and Sons, New York, NY, pp. 93–115.

Robinson, M. J. and Cobb, M. H. (1997) Mitogen-activated protein kinase pathways, Curr. Op. Cell. Biol. 9, 180–186.

The Receptor

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J. D. (eds.) (1994) Molecular Biology of the Cell, 3rd ed. Garland Publishing, NY, p. 755.

Amaya, E., Musci, T. J., and Kirschner, M. W. (1991) Expression of a dominant negative mutant of the FGF receptor disrupts mesoderm formation in Xenopus embryos, Cell 66, 257–270.

Chai, N, Patel, Y., Jacobson, K, McMahon, J., and Rappolee, D. A., (1998) FGF is an essential positive regulator of the fifth cell division in all stem cells of the preimplantation mouse embryo, Devel. Biol. 198, 105–115.

Clyman, R. I., Peters, K. G., Chen, Y. Q., Escobedo, J., Williams, L. T., Ives, H. E., and Wilson, E. (1994) Phospholipase C gamma activation, phosphotidylinositol hydrolysis, and calcium mobilization are not required for FGF receptor-mediated chemotaxis, Cell Adhes. Commun. 4, 333–342.

Feldman, B., Poueymirou, W., Papaioannou, V. E., DeChiara, T. M., and Goldfarb, M. (1995) Requirement of FGF-4 for postimplantation mouse development, Science 267, 246–249.

MacNicol, A. M., Muslin, A. J., and Williams, L. T. (1993) Raf-1 kinase is essential for early Xenopus development and mediates the induction of mesoderm by FGF, Cell 73, 571–583

Ornitz, D. M., Xu, J., Colvin, J. S., McEwen, D. G., MacArthur, C. A., Coulier, F., Gao, G., and Goldarb, M. (1996) Receptor specificity of the fibroblast growth factor family, J. Biol. Chem. 271, 15,292–15,297.

Rappolee, D. A. (1998) Growth factors and hormones in mammalian development. Growth factors in the mammalian pre- and post-implantation embryo, In: Hormones and Growth Factors in Development and Neoplasia, (Dickson, R. and Salomon, D., eds.) John Wiley and Sons, New York, NY, pp. 93–115.

Sorokin, A., Mohammadi, M., Huang, J., and Schlessinger, J. (1994). Internalization of fibroblast growth factor receptor is inhibited by a point mutation at tyrosine 766, J. Biol. Chem. 269, 17,056–17,061.

Ueno, H., Gunn, M., Dell, K., Tseng, Jr., A., and Williams, L. (1992) A truncated form of fibroblast growth factor receptor 1 inhibits signal-transduction by multiple types of fibroblast growth factor receptor, J. Biol. Chem. 267, 1470–1476.

The Signal Cascade

Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K., Watson, J. D. (eds.) (1994) Molecular Biology of the Cell, 3rd ed., Garland Publishing, NY, p. 755.

Hill, C. S. and Treisman, R. (1999) Growth Factors and Gene Expression: Fresh Insights from Arrays, Science STKE 3, 1–6.

Feramisco, J. R., Clark, R., Wong, G., Arnheim, N., Milley, R., and McCormick, F. (1985) Transient reversion of *ras* oncogene-induced cell transformation by antibodies specific for amino acid 12 of *ras* protein, Nature 314, 639–642.

Jun, T., Gjoerup, O., and Roberts, T. M. (1999) Tangled Webs: Evidence of Cross-Talk Between c-Raf-1 and Akt, Science STKE 13, 1–3.

Turning Off the Response

Shaw, G. and Kamen, R. (1986) A conserved AU sequence from the 3' untranslated region of GM-CSF mRNA mediates selective mRNA degradation, Cell 46, 659–667.

Sun, H., Charles, C. H., Lau, L. F., and Tonks, N. K. (1993) MKP-1 (3CH134), an immediate early gene product is a dual specificity phosphatase that dephosphorylates MAP kinase in vivo, Cell 75, 487–493.

Receptor Tyrosine Kinases (FGF Receptor) and Mitogenesis

Dekker, L. V. and Parker, P. J. (1994) Protein kinase C-a question of specificity, Trends Biosci. 19, 73–77.

Deng, C., Wynshaw-Boris, A., Zhou, F., Kuo, A., and Leder, P. (1996) Fibroblast growth factor receptor 3 is a negative regulator of bone growth, Cell 84, 911–921.

Huang, J., Mohammadi, M., Rodrigues, G. A., and Schlessinger, J. (1995) Reduced activation of RAF-1 and MAP kinase by a fibroblast growth factor receptor mutant deficient in stimulation of phosphatidylinositol hydrolysis, J. Biol. Chem. 270, 5065–5072.

Landgren, E., Blume-Jensen, P., Courtneidge, S. A., and Claesson-Welsh, L. (1995) Fibroblast growth factor receptor-1 regulation of Src family kinases. Oncogene 10, 2027–2035.

LaVoie, J. N., L'Alemain, G., Brunet, A., Muller, R., and Pouyssegur, J. (1996) Cyclin D1 expression is regulated positively by the p42/p44 MAPK and negatively regulated by the p38/HOGMAPK pathway, J. Biol. Chem. 271, 20,608–20,616.

Mohammadi, M., Dionne, C. A., Li, W., Li, N., Spivak, T., Honegger, A. M., Jaye, M., and Schlessinger, J. (1992) Point mutation in FGF receptor eliminates phosphatidylinositol hydrolysis without affecting mitogenesis, Nature 358, 681–684.

Mohammadi, M., Dikic, I., Sorokin, A., Burgess, W., Jaye, M., and Schlessinger, J. (1996) Identification of six novel autophosporylation sites on fibroblast growth factor receptor-1 and elucidation of their importance in receptor activation and signal-transduction, Mol. Cell. Biol. 16, 977–989.

Newton, A. C. (1997) Regulation of protein kinase C, Curr. Opin. Cell. Biol. 9, 161–169.

Partanen, J., Vainikka, S., Korhonen, J., Armstrong, E., and Alitalo, K. (1992) Diverse receptors for FGFs, Prog. Growth Factor Res. 4, 69–83.

Peters, K. G., Marie, J., Wilson, E., Ives, H. E., Escobedo, J., Del Rosario, M., Mirda, D., and Williams, L. T. (1992) Point mutation of an FGF receptor abolishes phosphatidylinositol turnover and Ca2+ flux but not mitogenesis, Nature 358, 678–681.

Sahni, M., Ambrosetti, D. C., Mansukhani, A., Gertner, R., Levy, D., and Basilico, C. (1999) FGF signaling inhibits chondrocyte proliferation and regulates bone development through the STAT-1 pathway, Genes Dev. 13, 1361–1366.

Spivak-Kroizman, T., Mohammadi, M., Hu, P., Jaye, M., Schlessinger, J., and Lax, I. (1994) Point mutation in the fibroblast growth factor receptor eliminates phosphatidylinositol hydrolysis without affecting neuronal differentiation of PC12 cells, J. Biol. Chem. 269, 14,419–14,423.

Transcription Factors

Amati, B., Dalton, S., Brooks, M. W., Littlewood, T. D., Evan, G. I., and Land, H. (1992) Transcriptional activation by the human c-Myc oncoprotein in yeast requires interaction with Max, Nature 359, 423–426.

Binétruy, B., Smeal, T., and Karin, M. (1991) Ha-Ras augments c-jun activity and stimulates phosphorylation of its activation domain, Nature 351, 122–127.

Franza, B. R., Rauscher, F. J., Josephs, S. F., and Curran, T. (1988). The Fos complex and Fos related antigens recognize sequence elements that contain AP-1 binding sites, Science 239, 1150–1153.

Pulverer, B. J., Kyriakis, J. M., Avruch, J., Nikolakaki, E., and Woodgett, J. R. (1991) Phosphorylation of c-Jun mediated by MAP kinases, Nature 353, 670–674.

ATF

Gupta, S., Campbell, D., Derijard, B., and Davis, R. J. (1995) Transcription factor ATF2 regulation by the JNK signal transduction pathway, Science 267, 389–393.

Livingstone, C., Patel, G., and Jones, N. (1995) ATF-2 contains a phosphorylation-dependent transcriptional activation domain. EMBO J. 14, 1785–1797.

Montminy, M. R., Sevarino, K. A., Wagner, J. A., Mandel, G., and Goodman, R. H. (1986) Identification of a cyclic-AMP-responsive element within the rat somatostatin gene, Proc. Natl. Acad. Sci. USA 83, 6682–6686.

ERK (MAPK Family)

Bar-Sagi, D. and Feramisco, J. R. (1986) Induction of membrane ruffling and fluid-phase pinocytosis in quiescent fibroblasts by *ras* proteins, Science 233, 1061–1068.

Boulton, T. G., Nye, S. H., Robbins, D. J., Ip, N. Y., Radziejewska, E., Morgenbesser, S. D., et al. (1991) ERKs: a family of protein-serine/threonine kinases that are activated and tyrosine phosphorylated in response to insulin and NGF, Cell 65, 663–675.

Cheng, A. M., Saxton, T. M., Sakai, R., Kulkarni, S., Mbalmalu, G., Vogel, W., et al. (1998) Mammalian Grb2 regulates multiple steps in embryonic development and malignant transformation, Cell 95, 793–803.

Crews, C. M., Alessandrini, A., and Erikson, R. L. (1992) The primary structure of MEK, a protein kinase that phosphorylates the ERK gene product, Science 258, 478–480.

Dérijard, B., Raingeaud, J., Barrett, T., Wu, I.-H., Han, J., Ulevitch, R. J., and Davis, R. J. (1995) Independent human MAP-kinase signal-transduction pathways defined by MEK and MKK isoforms, Science 267, 682–685.

Diaz-Meco, M. T., Berra, E., Municio, M. M., Sanz, L., Lozano, J., Dominguez, I., et al. (1993) A dominant negative protein kinase C β subspecies blocks NF-kappa B activation, Mol. Cell. Biol. 13, 4770–4775.

Kouhara, H., Hadari, Y. R., Spivak-Kroizman, T., Schilling, J., Bar-Sagi, D., Lax, I., and Schlessinger, J. (1997) A lipid-anchored Grb2-binding protein that links FGF-receptor activation to the Ras/MAPK signaling pathway, Cell 89, 693–702.

LaMorte, V. J., Kennedy, E. D., Collins, L. R., Goldstein, D., Harootunian, A. T., Brown, J. H., and Feramisco, J. R. (1993) A requirement for Ras protein function in thrombin-stimulated mitogenesis in astrocytoma cells, J. Biol. Chem. 268, 19,411–19,415.

Lechner, C., Zahalka, M. A., Giot, J. F., Moller, N. P., and Ullrich, A. (1996) ERK6, a mitogen-activated protein kinase involved in C2C12 myoblast differentiation, Proc. Natl. Acad. Sci. USA 93, 4355–4359.

Morrison, D. K., Kaplan, D. R., Rapp, U., and Roberts, T. M. (1988) Signal transduction from membrane to cytoplasm: growth factors and membrane-bound oncogene products increase Raf-1 phosphorylation and associated protein kinase activity, Proc. Natl. Acad. Sci. USA 85, 8855–8859.

Payne, D. M., Rossomando, A. J., Martino, P., Erickson, A. K., Her, J. H., Shabanowitz, J., et al. (1991) Identification of the regulatory phosphorylation sites in pp42/mitogen-activated protein kinase (MAP kinase), EMBO J. 10, 885–892.

Sahai, E., Olson, M. F., and Marshall, C. J. (2001) Cross-talk between *ras* and Rho signaling pathways in transformation favours proliferation and increased motility, EMBO J. 20, 755–766.

Williams, N. G., Roberts, T. M., and Li, P. (1992) Both p21 *ras* and pp60 v-*src* are required, but neither alone is sufficient, to activate the Raf-1 kinase. Proc. Natl. Acad. Sci. USA 89, 2922–2926.

Zhou, G., Bao, Z. Q., and Dixon, J. E. (1995) Components of a new human protein kinase signal transduction pathway, J. Biol. Chem. 270, 12,665–12,669.

Grbs SHc SHb

Cheng, A. M., Saxton, T. M., Sakai, R., Kulkarni, S., Mbalmalu, G., Vogel, W., et al. (1998) Mammalian Grb2 regulates multiple steps in embryonic development and malignant transformation, Cell 95, 793–803.

GTPase Releasing Factors (GRF), SOS Enhancing Factors (GEF), and Activating Factors (GAP)

Cherfils, J. and Chardon, P. (1999) GEFs: Structural basis for their activation of GTP-binding proteins, Trends Biochem. Sci. 24, 306–311.

IRS1-4, FRS-2

Myers, M. G., Wang, L.-M., Sun, X. J., Zhang, Y., Yenush, L., Schlessinger, J., Pierce, J. H., and White, M. F. (1994) Role of IRS-1-GRB-2 complexes in insulin signaling, Mol. Cell. Biol. 14, 3577–3587.

MAPKAP

Waskiewicz, A. J., Flynn, A., Proud, C. G., and Cooper, J. A. (1997) Mitogen-activated protein kinases activate the serine/threonine kinases Mnk1 and Mnk2, EMBO J. 16, 1090–1920.

MEK

Crews, C. M., Alessandrini, A., and Erikson, R. L. (1992) The primary structure of MEK, a protein kinase that phosphorylates the ERK gene product, Science 258, 478–480.

Holland, P. M., Suzanne, M., Campbell, J. S., Noselli, S., and Cooper, J. A. (1997) MKK7 is a stress-activated mitogen-activated protein kinase kinase functionally related to hemipterous, J. Biol. Chem. 272:24,994–24,998.

Tournier, C., Whitmarsh, A. J., Cavanagh, J., Barrett, T., and Davis, R. J. (1997) Mitogen-activated protein kinase kinase 7 is an activator of the c-Jun NH2-terminal kinase, Proc. Natl. Acad. Sci. USA 94, 7337–7342.

Wu, J., Harrison, J. K., Dent, P., Lynch, K. R., Weber, M. J., and Sturgill, T.W. (1997) Identification and characterization of a new mammalian mitogen-activated protein kinase kinase, MKK2, Mol. Cell. Biol. 13, 4539–4548.

MSK-1

Kozma, S. C., Ferrari, S., Bassand, P., Siegmann, M., Totty, N., and Thomas, G. (1990) Cloning of the mitogen-activated S6 kinase from rat liver reveals an enzyme of the second messenger subfamily, Proc. Natl. Acad. Sci. USA 87, 7365–7369.

Moller, D. E., Xia, C. H., Tang, W., Zhu, A.X., and Jakubowski, M. (1994) Human rsk isoforms: cloning and characterization of tissue-specific expression, Am. J. Physiol. 266, C351–C359.

Zhao, Y., Bjorbaek, C., Weremowicz, S., Morton, C. C., and Moller, D. E. (1995) RSK3 encodes a novel pp90rsk isoform with a unique N-terminal sequence: growth factor-stimulated kinase function and nuclear translocation, Mol. Cell. Biol. 15, 4353–4363.

p38 (MAPK Family)

Alam, J., Wicks, C., Stewart, D., Gong, P., Touchard, C., Otterbein, S., et al. (2000) Mechanism of heme oxygenase-1 gene activation by cadmium in MCF-7 mammary epithelial cells. Role of p38 kinase and Nrf2 transcription factor, J. Biol. Chem. 275, 27,694–27,702.

Brewster, J. L., de Valoir, T., Dwyer, N. D., Winter, E., and Gustin, M. C. (1993) An osmosensing signal transduction pathway in yeast, Science 259, 1760–1763.

Goedert, M., Cuenda, A., Craxton, M., Jakes, R., and Cohen, P. (1997) Activation of the novel stress-activated protein kinase SAPK4 by cytokines and cellular stresses is mediated by SKK3 (MKK6); comparison of its substrate specificity with that of other SAP kinases, EMBO J. 16, 3563–3571.

Han, J., Lee, J.-D., Bibbs, L., and Ulevitch, R. J. (1994) A MAP kinase targeted by endotoxin and hyperosmolarity in mammalian cells, Science 265, 808–811.

Jiang, Y., Chen, C., Li, Z., Guo, W., Gegner, J. A, Lin, S., and Han, J. (1996) Characterization of the structure and function of a new mitogen-activated protein kinase (p38beta), J. Biol. Chem. 271, 17,920–17,926.

Nishida, E. and Gotoh, Y. (1993) The MAP kinase cascade is essential for diverse signal transduction pathways, Trends Biochem. Sci. 18, 128–131.

Olofsson, B., Chardin, P., Toudrot, N., Zahraoni, A., and Tavitian, A. (1988) Expression of the *ras*-related ral A rho 12 and rab genes in adult mouse tissues, Oncogene 3, 231–234.

Raingeaud, J., Gupta, S., Rogers, J. S., Dickens, M., Han, J., Ulevitch, R. J., and Davis, R. J. (1995) Pro-inflammatory cytokines and environmental stress cause p38 mitogen-activated protein kinase activation by dual phosphorylation on tyrosine and threonine, J. Biol. Chem. 270, 7420–7426.

Schonwasser, D. C., Marais, R. M., Marshall, C. J., and Parker, P. J. (1998) Activation of the mitogen-activated protein kinase/extracellular signa-regulated kinase pathway by conventional, novel, and atypical protein kinase C isotypes, Mol. Cell Biol. 18, 790–798.

Stacey, D. W., Feig, L. A., and Gibbs, J. B. (1991) Dominant inhibitory Ras mutants selectively inhibit the activity of either cellular or oncogenic Ras, Mol. Cell. Biol. 11, 4053–64.

Ueda, Y. Hirsi, S., Osada, S., Atsushi, S., Mizuno, K., and Ohno, S. (1996) Protein kinase C activates the MEK-ERK pathway in a manner independent of *ras* and dependent on raf, J. Biol. Chem. 271, 23,512–23,519.

Wang, X. S., Diener, K., Manthey, C. L., Wang, S. W., Rosenweig, B., Bray, J., et al. (1997) Molecular cloning and characterization of a novel p38 mitogen activated protein kinase, J. Biol. Chem. 272, 23668–23674.

Zervos, A. S., Faccio, L., Gatto, J. P., Kyriakis, J. M., and Brent, R. (1995) Mxi2, a mitogen-activated protein kinase that recognizes and phosphorylates Max protein. Proc. Natl. Acad. Sci. USA 92, 10,531–10,534.

Raf Family (aka MAPKKK)

Avruch, J., Zhang, X. F., and Kyriakis, J. M. (1994) Raf meets Ras: completing the framework of a signal transduction pathway, Trends Biochem. Sci. 19, 279–283.

Downward, J. (1995) KSR: a novel player in the RAS pathway, Cell. 83, 831–834.

Heidecker, G., Huleihel, M., Cleveland, J. L., Kolch, W., Beck, T. W., Lloyd, P., et al. (1990) Mutational activation of c-raf-1 and definition of the minimal transforming sequence, Mol. Cell. Biol. 10, 2503–2512.

Morrison, D. K., Kaplan, D. R., Rapp, U., and Roberts, T. M. (1988). Signal transduction from membrane to cytoplasm: growth factors and membrane-bound oncogene products increase Raf-1 phosphorylation and associated protein kinase activity, Proc. Natl. Acad. Sci. USA 85, 8855–8859.

Turner, B., Rapp, U., App, H., Greene, M., Dobashi, K., and Reed, J. (1997) Interleukin 2 induces tyrosine phosphorylation and activation of p72-74 Raf-1 kinase in a T-cell line, Proc. Natl. Acad. Sci. USA 88, 1227–1231.

Williams, N. G., Roberts, T. M., and Li, P. (1992) Both p21ras and pp60v-src are required, but neither alone is sufficient, to activate the Raf-1 kinase, Proc. Natl. Acad. Sci. USA 89, 2922–2926.

Yamaguchi, K., Shirakabe, K., Shibuya, H., Irie, K., Oishi, I., Ueno, N., et al. (1995) Identification of a member of the MAPKKK family as a potential mediator of TGF-β signal transduction, Science 270, 2008–2011.

Ras-G Proteins (ras-Rac-Rho-Rap1,2-Ral)

Olofsson, B. (1997) Rho guanine dissociation inhibitors: pivotal molecules in cellular signalling, Cell Signal. 11, 545–554.

Sahai, E., Olson, M. F., and Marshall, C. J. (2001) Cross-talk between *ras* and Rho signaling pathways in transformation favours proliferation and increased motility, EMBO J. 20, 755–766.

RSK

Bjorbaek, C., Zhao, Y., and Moller, D. E. (1995) Divergent functional roles for p90rsk kinase domains, J. Biol. Chem. 270, 18,848–18,852.

Jensen, C. J., Buch, M. B., Krag, T.O., Hemmings, B. A., Gammeltoft, S., and Frodin, M. (1999) 90-kDa ribosomal S6 kinase is phosphorylated and activated by 3-phosphoinositide-dependent protein kinase-1, J. Biol. Chem. 274, 27,168–27,176.

Kozma, S. C., Ferrari, S., Bassand, P., Siegmann, M., Totty, N., and Thomas, G. (1990) Cloning of the mitogen-activated S6 kinase from rat liver reveals an enzyme of the second messenger subfamily, Proc. Natl. Acad. Sci. USA 87, 7365–7369.

Moller, D. E., Xia, C.H., Tang, W., Zhu, A. X., and Jakubowski, M. (1994) Human rsk isoforms: cloning and characterization of tissue-specific expression, Am. J. Physiol. 266, C351–C359.

Zhao, Y., Bjorbaek, C., Weremowicz, S., Morton, C. C., and Moller, D. E. (1995) RSK3 encodes a novel pp90rsk isoform with a unique N-terminal sequence: growth factor-stimulated kinase function and nuclear translocation, Mol. Cell Biol. 15, 4353–4363.

SAPK/JNK (MAPK Family)

Derijard, B., Hibi, M., Wu, I.-H., Barrett, T., Su, B., Deng, T., Karin, M., and Davis, R. J. (1994) JNK1: a protein kinase stimulated by UV light and Ha-Ras that binds and phosphorylates the c-Jun activation domain, Cell 76, 1025–1037.

Devary, Y., Rosette, C., DiDonato, J. A., and Karin, M. (1993). NF-κ B activation by ultraviolet light not dependent on a nuclear signal, Science 261, 1442–1445.

Waskiewicz, A. J., Flynn, A., Proud, C. G., and Cooper, J. A. (1997) Mitogen-actiated protein kinases activate the serine/threonine kinases Mnk1 and Mnk2, EMBO J. 16, 1090–1920.

Yamaguchi, K., Shirakabe, K., Shibuya, H., Irie, K., Oishi, I., Ueno, N., et al. (1995) Identification of a member of the MAPKKK family as a potential mediator of TGF-β signal-transduction, Science 270, 2008–2011.

STAT

Sahni, M., Ambrosetti, D. C., Mansukhani, A., Gertner, R., Levy, D., and Basilico, C. (1999) FGF signaling inhibits chondrocyte proliferation and regulates bone development through the STAT-1 pathway. Genes Dev. 13, 1361–1366.

B. Transcription and Translation

4 DNA Replication, Repair, and Recombination

Linda B. Bloom

Introduction

DNA contains the instructions needed to produce and regulate the components of a cell. This information is encoded in the order, or sequence, of the four possible nucleotide bases contained in the DNA polymer. Hydrogen bonding interactions between complementary base pairs bring two linear DNA polymers together to form a double helix. James Watson and Francis Crick not only deduced the double helical structure of DNA but also recognized that hydrogen bonding between complementary DNA bases would provide a mechanism for DNA duplication. Watson-Crick base pairing allows one strand of DNA to serve as a template for synthesis of a new strand by directing the incorporation of adenine opposite thymine and guanine opposite cytosine (*see* Fig. 1). This complementary base pairing also provides a mechanism for preserving the code. If one strand of DNA is damaged, the other can be used as a template to regenerate the damaged strand and recover information encoded in its sequence.

DNA Replication

Overview of DNA Replication

Every time a cell divides, its DNA must be duplicated so that each daughter cell receives an identical copy of instructions. The size of a DNA molecule alone makes replication an amazing undertaking. For example, human cells contain 3 billion base pairs of DNA divided into chromosomes ranging in size from about 50–250 million base pairs. If they were completely stretched out, these DNA molecules would range in length from about 1.7–8.5 cm. Many enzymes and proteins are required to physically manipulate these large polymers and to catalyze the synthesis of new DNA. These enzymes and the process of DNA replication are regulated by the cell so that replication is complete and genomes are duplicated only once every cell division.

DNA replication begins at a specific time in the cell cycle and at specific sites, *origins of replication*, in the genome. The DNA duplex is unwound at these replication origins to allow the enzymes that synthesize DNA access to the individual DNA strands (*see* Fig. 2). Each strand of parental DNA serves as a template for a *DNA polymerase* to make a new strand of DNA. Single nucleotide monomers that form Watson-Crick pairs with template bases are incorporated into a new DNA polymer by

See companion CD for color Fig. 2

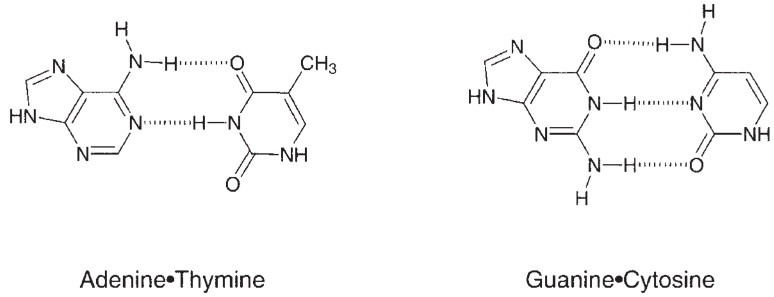


Fig. 1. Watson-Crick A•T and G•C base pairs.

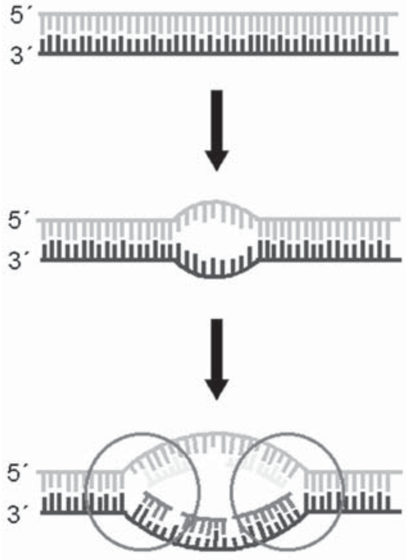


Fig. 2. Unwinding DNA at the origin of replication and the formation of replication forks. DNA replication begins at specific sites known as origins of replication. Origin binding proteins recognize these sites and initiate unwinding of the DNA duplex so that replication proteins can access the individual strands of DNA. Initially a small bubble is formed that is opened further by the activity of a DNA helicase. Replication complexes assemble on both sides of the bubble and these replication forks (circled) move away from the origin in both directions so that replication is bidirectional. At each fork, two new copies of DNA are synthesized using the parental strands as a templates.

DNA polymerases. As the new DNA grows, the parental duplex is progressively unwound forming *replication forks* that move away from the origin. DNA replication is *semi-conservative*, ultimately forming two DNA double helices that contain one strand of parental DNA and one strand of new DNA. In bacteria, replication forks move at a rate of about 500 nucleotides per second while in eukaryotes they move somewhat slower. Because genomes of eukaryotes are in general larger than bacteria and replication fork movement is slower, replication in eukaryotes is initiated at several origins instead of a single origin so that DNA duplication can be accomplished in a reasonable amount of time.

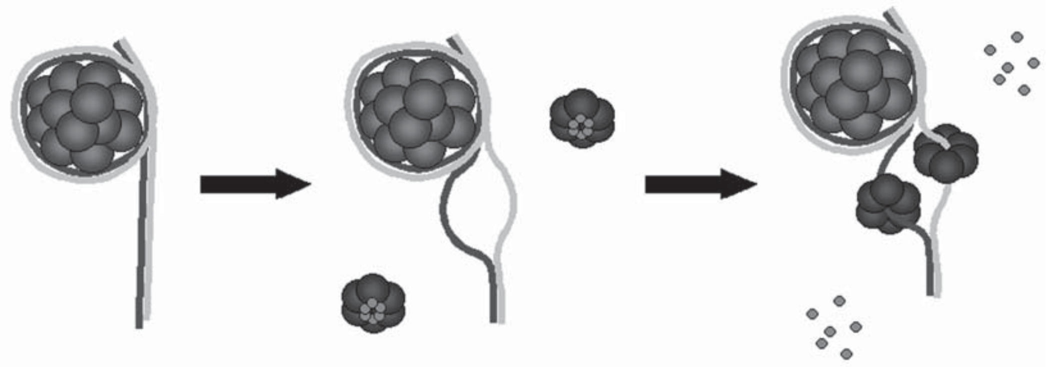


Fig. 3. Initiation of DNA replication at *oriC* in *E. coli*. DnaA protein binds to *oriC* to form a protein-DNA complex where the DNA is wrapped around several molecules of DnaA protein. DnaA binding induces unwinding of the DNA duplex at the A•T rich segments. DnaC protein binds the ring-shaped hexameric DnaB helicase and assembles the helicase onto the origin. One helicase complex is assembled at each fork of the replication bubble. After assembling the helicase, DnaC is released.

Much of what we know about DNA replication is based on studies in bacteria and viruses. However, more recent investigations of eukaryotic systems are revealing that many of the features of the bacterial replication machinery are also common to eukaryotes. Because replication in bacteria has been studied in greater detail to date, the *Escherichia coli* replication machinery will be presented for illustrative purposes and compared to eukaryotic systems.

Initiation of DNA Replication

Origins of DNA replication contain two general DNA sequence elements, an element that is relatively easy to unwind, and an element that is recognized by initiation proteins. The genome of *E. coli* exists as a single circular DNA molecule of about 4.5 million base pairs and contains a single 245 base pair (bp) origin of replication, *oriC*. Within *oriC* are three 13-bp AT-rich regions of DNA that are relatively easy to unwind and four 9-bp regions that are recognized by the *E. coli* initiator protein, DnaA. A complex containing several DnaA molecules binds to *oriC* in the region containing the 9-bp repeats and bends the DNA. This bending helps to unwind the DNA helix at the 13-bp AT-rich sequences (*see* Fig. 3). Many of the steps in DNA replication and repair, such as the unwinding of the origin, require energy to manipulate the structures of macromolecules and disrupt noncovalent interactions such as hydrogen bonding. The enzymes catalyzing these changes utilize the chemical energy stored in the phosphate bonds of adenosine-5'-triphosphate (ATP) to do the mechanical work. The DnaA protein utilizes the energy gained from ATP hydrolysis to power the unwinding of DNA at the origin. An origin recognition complex also exists in eukaryotes but its mechanism of action has not yet been completely defined.

See companion CD for color Fig. 3

Once the DNA duplex is opened, a *DNA helicase* can be loaded onto the single-stranded DNA to continue the unwinding process. DNA helicases are enzymes that utilize the energy from hydrolysis of ribonucleoside 5-triphosphates, most commonly ATP, to break hydrogen bonding interactions between complementary DNA strands and unwind nucleic acid duplexes. In *E. coli*, six molecules of the DnaB protein form a ring-shaped hexamer that encircles single-stranded DNA and functions as a helicase. This hexameric helicase structure is common to other organisms including eukaryotes where the MCM (Mini-Chromosome Maintenance) proteins are believed to perform the function of replicative DNA helicase. In *E. coli*, the DnaB hexamer is assembled around DNA by the ATP-dependent activity of DnaC (*see* Fig. 3).

Before DNA synthesis can begin, RNA primers must be made. DNA polymerases are unable to synthesize DNA *de novo* and can only extend RNA (or DNA) primers that are already paired with the template to be copied. *Primases* synthesize these primers using ribonucleoside 5'-triphosphates as building blocks to form a short strand of RNA complementary to the DNA template. The *E. coli* primase interacts with the DnaB helicase and begins synthesis of RNA primers shortly after DnaB has begun to unwind DNA. In eukaryotes, a hybrid RNA-DNA primer is synthesized by an enzyme complex containing both primase and DNA polymerase α. Once primers are formed, they can be extended by a DNA polymerase.

Enzymes at the Replication Fork

See companion CD for color Fig. 4

Assembly of a *replisome* is complete when the replicative DNA polymerase and its accessory proteins join the helicase and primase at the replication fork (*see* Fig. 4). The replisome will then continue to synthesize new DNA, unwinding the parental duplex as it goes. The actual synthesis of new DNA is catalyzed by a DNA polymerase contained within the replisome. Cells contain many different DNA polymerases that have different functions in DNA replication and repair. There are 5 known DNA polymerases in *E. coli* and at least a dozen in humans. In *E. coli*, DNA polymerase III catalyzes the bulk of DNA synthesis during replication and DNA polymerase δ does so in eukaryotes.

See companion CD for color Fig. 5

All DNA polymerases use 2'-deoxyribonucleoside–5'-triphosphates (dNTPs) as monomeric building blocks for making DNA. They catalyze the attack of the 3' hydroxyl group of the nucleotide at the primer end on the α-phosphoryl group of an incoming dNTP displacing pyrophosphate (*see* Fig. 5, upper panel). Thus, DNA polymerases extend DNA polymers in the 5' to 3' direction by incorporation of 2'-deoxyribonucleoside monophosphates. Watson-Crick base pairing interactions between the incoming dNTP and the next unpaired template base direct incorporation of correct nucleotides. Frequencies of adding incorrect nucleotides can be as low as one in a million, but even with this low error frequency mistakes will be made. To further reduce error frequencies, the DNA polymerases that function in replication contain a 3' to 5' exonuclease activity that allows them to *proofread* nucleotides that have been incorporated. This exonuclease activity catalyzes the hydrolysis of phosphodiester bonds to remove the last nucleotide added to the 3' primer end (*see* Fig. 5, lower panel). Thus, a nucleotide that has been added incorrectly can be removed.

The overall efficiency of synthesis by DNA polymerases is enhanced by accessory proteins which increase DNA polymerase *processivity*, or the number of nucleotides incorporated per DNA binding event. These accessory proteins consist of a ring-shaped sliding clamp that binds both DNA and the DNA polymerase and a clamp loader that assembles the clamp on DNA. Sliding clamps, made of identical protein subunits, encircle DNA and are capable of sliding along a DNA duplex (*see* Fig. 6).

See companion CD for color Fig. 6

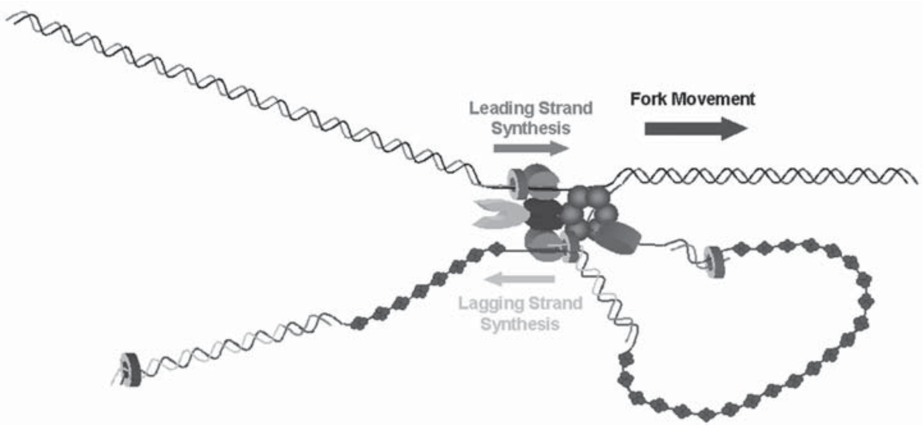


Fig. 4. Proteins at the *E. coli* replication fork.The dimeric polymerase complex is capable coordinated DNA synthesis on the leading and lagging strands. The leading strand polymerase synthesizes new DNA in the direction of fork movement and the lagging strand polymerase synthesizes DNA in the opposite direction. The hexameric helicase (light blue) unwinds DNA ahead of the polymerase and primase (red) makes RNA primers (red lines) on the lagging strand. Single-stranded DNA that forms as the helix unwinds is coated with single-stranded binding protein to prevent reannealing of strands and to remove secondary structure that may form within a single-strand. Sliding clamps (green) are assembled on each primer on the lagging strand by the clamp loading complex (yellow and dark blue). *See* companion CD for color.

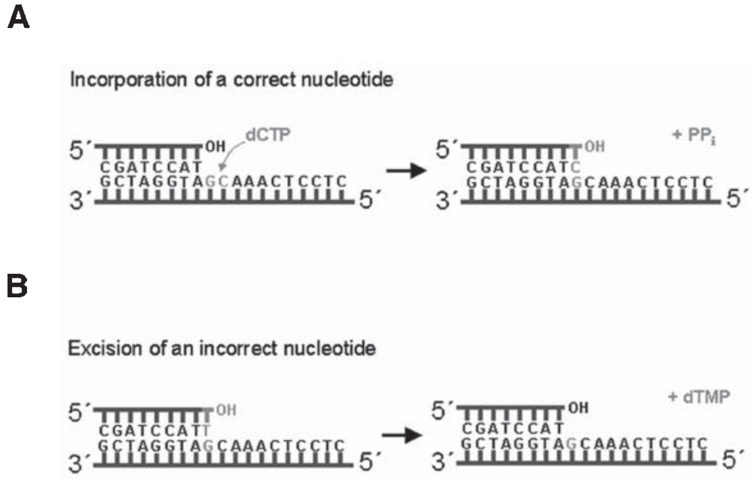


Fig. 5. Reactions catalyzed by DNA polymerases. **(A)** 2'-Deoxyribonucleoside 5'-triphosphates are used as substrates by DNA polymerases to extend a primer in template-directed reactions. The net reaction is incorporation of 2'-deoxyribonucleoside monophosphates onto the 3' hydroxyl of a primer with loss of pyrophosphate. **(B)** DNA polymerases can proofread newly incorporated nucleotides and excise incorrect nucleotides. The excision reaction removes the last nucleoside monophosphate that was incorporated.

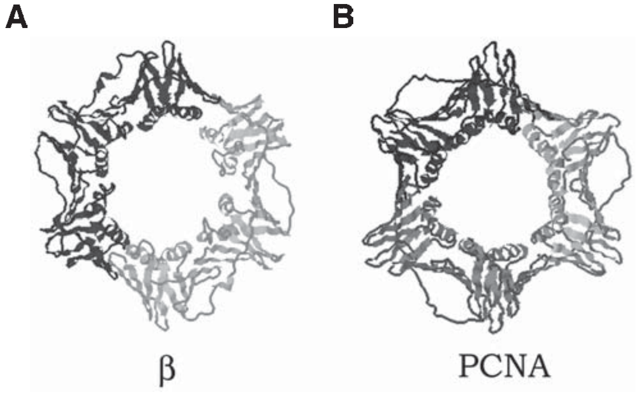


Fig. 6. Structures of sliding clamps from *E. coli* and humans. **(A)** The *E. coli* β sliding clamp is a head-to-tail dimer of identical monomer subunits. **(B)** The human PCNA sliding clamp is similar in overall structure to the β clamp but is composed of identical trimers. Each ring has a central hole that is large enough to encircle B-DNA.

By binding a sliding clamp, a DNA polymerase is effectively tethered to a DNA template so that it is capable of incorporating thousands of nucleotides without dissociating. In the absence of a sliding clamp, DNA synthesis is less efficient because DNA polymerases frequently dissociate from the template and must rebind to continue. Sliding clamps are assembled around DNA by the ATP-dependent activity of clamp loaders.

Leading and Lagging Strand Synthesis

In *E. coli*, a complex containing a dimeric DNA polymerase and accessory proteins interacts with the helicase and primase to form a replisome. This interaction stimulates the activity of the helicase and increases the rate of fork movement. The replisome, which contains two copies of DNA polymerase III, is capable of simultaneously copying both strands of parental DNA at the replication fork. But the two DNA polymerases must work in opposite directions to do this because DNA strands in a duplex are antiparallel and DNA polymerases can only synthesize DNA in the 5' to 3' direction. To accomplish this, one DNA polymerase working on the *leading strand*, synthesizes DNA in a single continuous piece moving in the direction of the replication fork. The other DNA polymerase working on the opposite or *lagging strand*, synthesizes DNA in shorter fragments named *Okazaki fragments* after Reiji Okazaki whose work led to their discovery (*see* Fig. 4). In *E. coli*, Okazaki fragments are 1000–2000 nt in length and in eukaryotes they are 100–200 nt. As the fork progresses, a loop of single-stranded DNA is created on the lagging strand and an RNA primer is synthesized by an enzyme called primase to begin each Okazaki fragment. The lagging strand polymerase extends these primers in the direction opposite to fork movement until it encounters a completed Okazaki fragment. Then, the polymerase dissociates and rebinds a new primer closer to the fork and extends it. Thus, the lagging strand polymerase must repeatedly dissociate from completed Okazaki fragments and rebind to new primers to continue DNA synthesis in a discontinuous manner. Overall, DNA synthesis is *semi-discontinuous* because it is made in one continuous strand on the leading strand and in discontinous fragments on the lagging strand.

To complete DNA replication, RNA primers must be replaced by DNA and Okazaki fragments must be joined together to form a continuous strand. In *E. coli*, removal of RNA primers and synthesis of DNA can be accomplished by a single enzyme, DNA polymerase I. The 5' to 3' exonuclease activity of DNA polymerase I degrades RNA primers while the 5' to 3' polymerase activity simultaneously synthesizes DNA to replace the RNA. In eukaryotes, separate enzymes are responsible for degrading the RNA and replacing it with DNA. Finally, DNA fragments on the lagging strand are joined by a *DNA ligase* to form one continuous polymer. DNA ligase catalyzes the formation of a phosphodiester bond between the 3' hydroxyl group at the end of one Okazaki fragment and the 5' phosphate at the beginning of the next. Any *nick* in one strand of a DNA duplex that has a 3' hydroxyl on one side and 5' phosphate on the other can be sealed by the activity of a DNA ligase.

Fidelity of DNA Replication and Mismatch Repair

DNA replication can be accomplished with as few as one mistake in a billion nucleotides incorporated. This amazing accuracy or *fidelity* of synthesis is achieved for the most part by the DNA polymerase but is enhanced by a group of mismatch repair enzymes that function to detect and correct replication errors. One main feature of a DNA polymerase that contributes to its fidelity is the geometry of the active site, which is optimized for binding Watson-Crick base pairs where the overall shape of both A•T and G•C pairs are the same (*see* Fig. 1). Mismatches such as G•T deviate from this ideal geometry so that incorrect nucleotides are incorporated much less efficiently. Frequencies of adding an incorrect nucleotide range from 1 in 1000 to 1 in 1,000,000 nucleotides depending on the nucleotide added. In the rare instant when a mistake is made, DNA polymerases have the ability to remove the incorrect nucleotide using the 3' to 5' exonuclease activity contained in the enzymes. This proofreading capability is further enhanced by a reduced efficiency of adding the next correct nucleotide onto a primer that ends with an incorrect nucleotide. Thus, when a mistake is made, the rate of adding more nucleotides is greatly reduced which allows the exonuclease time to remove the incorrect nucleotide. Once an incorrect nucleotide is removed, rapid incorporation of correct nucleotides by the DNA polymerase activity resumes. This proofreading activity increases the accuracy of DNA synthesis by a factor of about 10–100.

See companion CD for color Fig. 7

Mismatches that escape proofreading by the DNA polymerase can be corrected by the postreplicative mismatch repair process (*see* Fig. 7). The net result is the removal of a segment of DNA containing the incorrect nucleotide and resynthesis of DNA to replace the segment that was excised. The key enzymes responsible for mismatch repair in *E. coli* are MutS, MutL, MutH, and MutU. Mismatches are detected in double-stranded DNA by MutS which then interacts with the MutL protein. Together MutS and MutL signal where the mismatch is located to MutH and MutU. MutH is an endonuclease that is stimulated by MutL to cut the DNA strand containing the incorrect nucleotide. MutU is a DNA helicase that unwinds the duplex displacing the strand containing the incorrect nucleotide which is then degraded by an exonuclease. New DNA is synthesized to replace the segment that was removed. Homologs to MutS and MutL exist in eukaryotic cells and the overall repair process is similar.

How do the mismatch repair enzymes recognize which nucleotide of the mismatch is incorrect? In *E. coli*, the methylation status of the DNA allows the mismatch repair enzymes to distinguish between the newly synthesized DNA strand and the parental

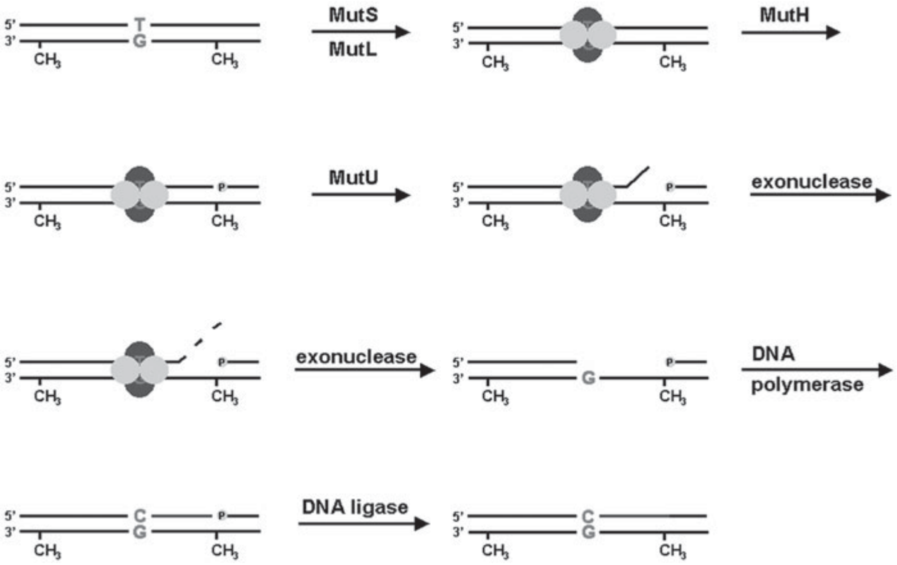


Fig. 7. Methyl-directed mismatch repair in *E. coli*. MutS protein recognizes and binds mismatches such as G•T in DNA and is joined by the MutL protein. MutL within the MutS-MutL-mismatched DNA complex stimulates the endonuclease activity of MutH to cleave the unmethylated DNA strand at the GATC sequence closest to the protein-mismatched DNA complex. The cut DNA strand is unwound by the activity of MutU helicase and then degraded by an exonuclease until the mismatch is removed. The missing segment of DNA is replaced by a DNA polymerase and the DNA strands are joined together by the activity of a DNA ligase. The letter P indicates a 5' phosphate group.

strand. Adenine is methylated in the *E. coli* genome when it appears in the sequence 5'GATC. This methylation of the genome occurs shortly after replication, so for a short time, the daughter strand is unmethylated while the parent strand is methylated. These 5'GATC sequences are also recognized by MutH which cuts the unmethylated daughter strand. These cut sites can be up to 1000–2000 nt away from the mismatch so a fairly large segment of DNA may be removed and replaced. While the overall process of mismatched repair is similar in eukaryotes, it is not yet clear how the eukaryotic enzymes distinguish between the newly synthesized strand and the parental strand.

DNA Recombination

Through the process of recombination, two DNA duplexes can exchange information to create hybrid molecules containing sequences from each of the original molecules. Recombination provides mechanisms for generating genetic diversity and for repairing DNA strand breaks. Recombination pathways can be grouped into two major classes, homologous and site-specific. Homologous recombination is the most general mechanism for recombination and plays a central role in both the generation of genetic diversity and repairing DNA. Homologous recombination occurs between two regions of DNA with similar or homologous sequences. Crosses between these two regions produce two new molecules that are hybrids of the original sequences.

See companion CD for color Fig. 8

During meiosis (process of cell division that ultimately produces germ cells containing a single copy of each chromosome), these crosses allow alleles (alternate forms of the same gene) to be exchanged between homologous chromosomes so that chromosomes passed onto haploid daughter cells are a hybrid of their progenitors (*see* Fig. 8A). This process allows a child to inherit traits from each of its grandparents even though the child only receives a single chromosome from each of its parents. Recombination between homologous duplexes also provides a mechanism for repairing a double-stranded break in one duplex or a damaged segment of DNA.

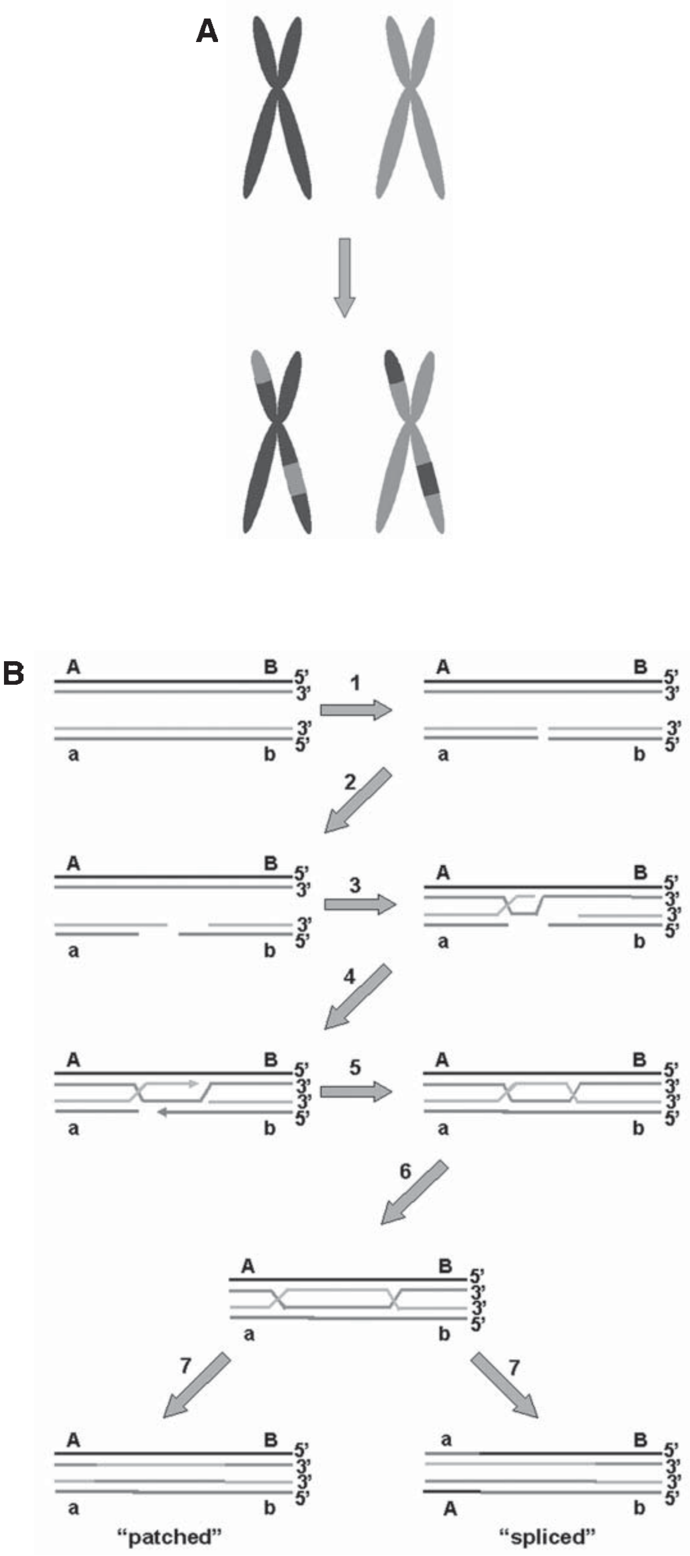
Site-specific recombination does not require homologous sequences between two DNA duplexes. As its name implies, site-specific recombination occurs when one DNA sequence is inserted into a specific site in another DNA duplex. This type of recombination produces a new DNA duplex where information from one is *spliced* into the other duplex. Some viruses use site-specific recombination to integrate their genomes into the genome of a host. Bacteriophage λ integrates its viral genome into the genome of its *E. coli* host and retroviruses such as HIV integrate a double-stranded DNA copy of their viral RNA genome into the host genome. Transposition is an example of site-specific recombination where a genetic element, transposon, moves from one location in a genome to another. This repositioning requires a specific nuclease, a transposase, encoded within the transposon. Site-specific recombination also provides a mechanism for immune cells to rearrange genetic elements to generate the diversity necessary to produce many different antibodies. By assembling different combinations of genetic elements through the process of V(D)J recombination immune cells are capable of producing genes encoding a multitude of different immunoglobulins that recognize different antigens.

Homologous Recombination

Homologous recombination is the most general pathway of recombination and is fairly well-defined. Homology or significant complementarity in DNA sequences is a prerequisite for homologous recombination between two duplexes. In addition, the physical exchange of information requires breaking and rejoining of the DNA molecules. Modern models of homologous recombination propose that recombination is initiated by the formation of a double-stranded break in one of the two duplexes. These breaks can be created by specific enzymes or can result from DNA damage. The broken duplex will serve as a substrate to initiate recombination after it is partially degraded by an exonuclease to generate free single-stranded ends. One of the single-stranded ends will invade the intact duplex and pair with the homologous region to produce a heteroduplex containing strands from two separate DNA duplexes (*see* Fig. 8B). The *strand invasion* and homologous pairing reactions are not spontaneous but require the activity of an enzyme, RecA protein in *E. coli* and its homolog, Rad51, in eukaryotes, to catalyze the reactions.

See companion CD for color Fig. 8

The strand invasion reaction generates a D-loop structure formed by the complementary strand from the intact duplex that was displaced. The invading strand can be extended by a DNA polymerase, which increases the size of the single-stranded region in the D-loop. This opens up a region of DNA homologous to the other single-stranded end of the broken DNA which then pairs with the D-loop. This end can also be extended by a DNA polymerase. Extension of both ends effectively replaces the segments that were removed by the exonuclease to generate single-stranded ends that initiate recombination.


A
B
A B 5' 3'
3' 5' a b
1
A B 5' 3'
3' 5' a b
2
A B 5' 3'
3' 5' a b
3
A B 5' 3' 3' 5'
a b
4
A B 5' 3' 3' 5'
a b
5
A B 5' 3' 3' 5'
a b
6
A B 5' 3' 3' 5'
a b
7
7
A B 5' 3' 3' 5'
a b
"patched"
a B 5' 3' 3' 5'
A b
"spliced"

Two crossover points between the two homologous duplexes are created following strand invasion by one half of the broken duplex and pairing between the displaced D-loop and the other half of the broken duplex. These crossed strands are named *Holliday junctions* after Robyn Holliday, who first proposed their occurrence in homologous recombination. Holliday junctions can migrate down the DNA duplexes *unzipping* the original duplexes while simultaneously *zipping* these strands together with new partners to extend heteroduplex regions of DNA. This process of *branch migration* can continue as long as homology between the regions of DNA is high.

Following branch migration, the two DNA duplexes are separated by cleaving the DNA strands at the Holliday junctions. This cleavage is catalyzed by a nuclease or resolvase that is specific for DNA in Holliday structures. Depending on which strands of the junctions are cleaved, two distinct products can be formed that contain different segments of DNA from the original duplex (*see* Fig. 8B). A spliced product can be formed where each end of the hybrid duplex is derived from one of the original duplexes. Alternatively, a patched product can be formed where the hybrid duplex contains a single-stranded segment from one duplex within a duplex derived from the other original duplex.

DNA Repair

DNA, like any other molecule, can spontaneously decompose with time, react with other chemicals (naturally present in a cell), or be damaged by UV or ionizing radiation. Damage to DNA can have many deleterious effects on the cell. Some types of damage alter DNA structure so that the sequence is misread by DNA polymerases causing it to incorporate erroneous bases and generate mutations. Others are so severe that DNA synthesis is blocked at the site of damage. DNA damage occurs with a frequency high enough that it would be lethal to a cell if it were not repaired. Recombination, as discussed earlier, provides a mechanism for repairing DNA strand breaks.

There are two general strategies that can be taken by a cell to fix DNA damage, direct reversal of the damage, and removal of a segment of DNA containing damage. Direct reversal of DNA damage relies on different enzymes capable of catalyzing different chemical reactions to undo the damage. There are two known examples of enzymes that repair DNA by direct reversal of damage. The majority of DNA damage is repaired using an excision/resynthesis strategy where a group of enzymes removes a

Fig. 8. *(opposite page)* Homologous recombination. **(A)** Recombination between sister chromatids during meosis results in exchange of information to generate two new chromatids that are hybrids of the originals. **(B)** Double-strand break model for homologous recombination. In this model, recombination is initiated by forming a double-stranded break (step 1) in one of the homologous duplexes. The broken DNA is then processed by partial degradation by an exonuclease to generate single-stranded DNA on the 3' ends (step 2). One 3' single-stranded end invades the homologous duplex forming a D-loop in the intact duplex (step 3). The invading 3' end is extended by a DNA polymerase enlarging the D-loop which can then pair with the remaining 3' single-stranded end (step 4). As the D-loop expands, it can displace the 5' end of the broken duplex which is then free to pair with the intact duplex (step 5). Branch migration enlarges the regions of heteroduplex DNA by unzipping the regions that were originally paired and zipping them onto the homologous duplex (step 6). Finally, the cross-over points or Holliday junctions are resolved by cleavage of the crossing strands (step 7). Two different products, patched and spliced, are formed depending on which of the crossed strands are cleaved.

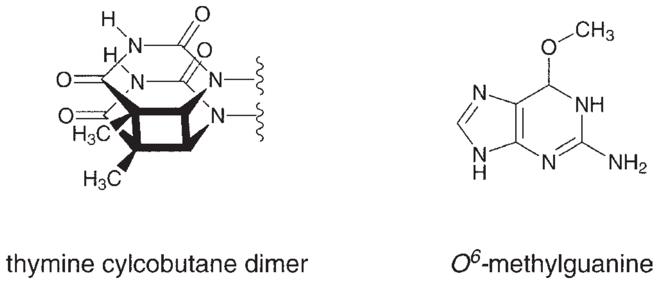


Fig. 9. Structures of a thymine cyclobutane dimer and O^6-methylguanine.

segment of DNA containing damage and replaces it with new DNA. For this type of process, the same basic set of enzymes can remove many types of DNA damage.

Direct Reversal of DNA Damage

DNA Photolyase Catalyzed Splitting of Thymine Cyclobutane Dimers

When exposed to UV light, two neighboring pyrimidine bases, C or T, in DNA can react with each other to become covalently joined. One reaction that occurs is the formation of thymine cyclobutane dimers (*see* Fig. 9). When these thymine dimers are formed, they distort the local structure of DNA and are no longer recognized as a pair of T's by DNA polymerases. In bacteria, the enzyme, DNA photolyase, reverses the formation of thymine dimers. Enzyme-bound cofactors absorb light and initiate an electron transfer reaction that catalyzes the splitting of the thymine dimer to regenerate two intact thymine bases.

Removal of Methyl Groups by O^6-Methylguanine Methyltransferase

Chemicals that are naturally present in the cell, as well as chemicals in the environment, can react with DNA to methylate bases. One product of this reaction, O^6-methylguanine (O^6-MeG), induces the incorporation of an incorrect base by a DNA polymerase (*see* Fig. 9). O^6-MeG is repaired by O^6-methylguanine methyltransferase (MGMT) by directly removing the methyl group to regenerate a normal guanine residue. MGMT is not an enzyme in the true sense of the word because it does not act catalytically; instead it is a suicide enzyme. The methyl group is transferred from guanine to a cysteine residue on the protein where a covalent bond is formed. Because MGMT cannot be demethylated, it is incapable of catalyzing the removal of other methyl groups and becomes inactive. MGMT is found in both bacteria and eukaryotes.

Repair by Excision/Resynthesis

The process of excision of a section of damaged DNA followed by resynthesis allows cells to use the same basic chemical reaction and a common set of enzymes to repair many different types of DNA damage. The two main pathways responsible for excision repair are *base excision repair* that usually replaces a single damaged base and *nucleotide excision repair* that replaces a short segment of DNA containing the damaged nucleotide.

Base Excision Repair

The base excision repair pathway primarily repairs damage to DNA bases much of which occurs spontaneously in the cell without the influence of environmental hazards. Examples include oxidation, deamination, and alkylation of DNA bases. These damaged bases are recognized by damage-specific *DNA glycosylases* that initiate the base excision repair pathway. DNA glycosylases bind to damaged bases and cleave the C1'-N glycosylic bond between the base and sugar. This leaves a baseless sugar, or *AP site* (for apurinic or apyrimidinic), in DNA that is removed by other enzymes in the pathway (*see* Fig. 10A). Several different DNA glycosylases are present in cells and each recognizes a specific damaged base or a class of damaged bases. For example, uracil DNA glycosylase recognizes and excises only uracil which can be formed in DNA by deamination of cytosine. Formamidopyrimidine (FaPy) DNA glycosylase recognizes several different bases damaged by oxidation.

See
companion CD
for color
Fig. 10

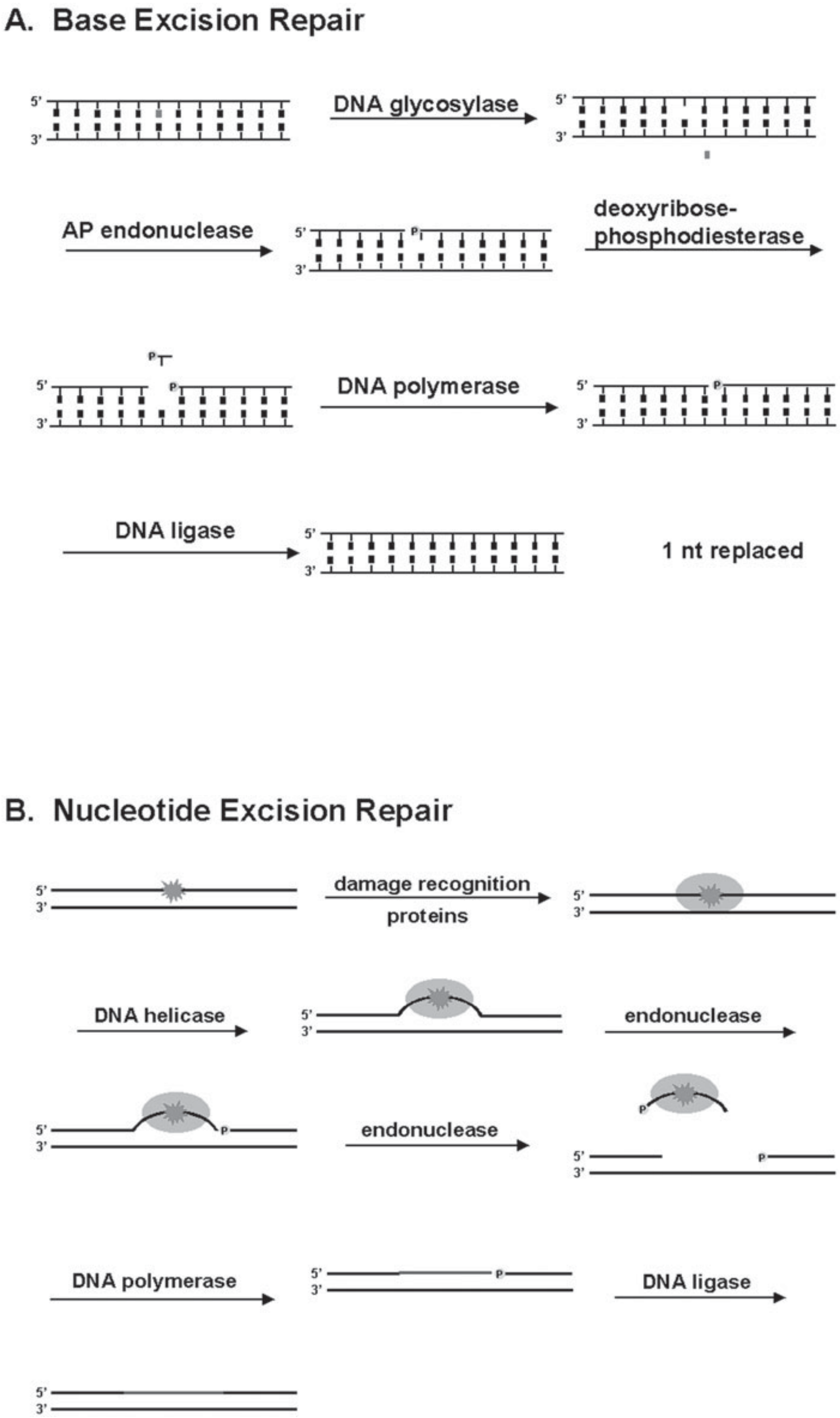
Once a DNA glycosylase has removed a damaged base, the AP site that it leaves must be repaired. An *AP endonuclease* starts this process by cleaving the phosphodiester bond on the 5' side of the AP site to generate a 3'hydroxyl and a deoxyribose phosphate. A deoxyribophosphodiesterase then cleaves the 3' phosphate of the baseless sugar to remove it and leave a one nucleotide gap in the DNA. A polymerase subsequently adds the missing nucleotide and the nick is sealed by the activity of a DNA ligase. In *E. coli*, separate enzymes catalyze the removal of the baseless sugar and the incorporation of the missing nucleotide. In humans, DNA polymerase β contains contains two enzymatic activities and is capable of both incorporating the missing nucleotide and removing the baseless sugar.

Nucleotide Excision Repair

The nucleotide excision repair pathway recognizes and repairs damage that generates larger more bulky lesions and local distortions in the DNA structure. In the nucleotide excision repair pathway, damage is recognized by a protein complex capable of identifying many different types of damage. It is believed that this complex recognizes distortions in the overall DNA structure at sites of damage or an increase in the ease of unwinding the duplex in the region of the damage. Regardless of the mechanism, this complex identifies sites of damage and helps recruit the rest of the repair machinery (*see* Fig. 10B). This repair complex unwinds the DNA duplex at the damaged site. Endonuclease activities within the complex cleave the DNA backbone both 5' and 3' to the site of damage. This creates a short oligonucleotide segment, 12–13 nt long in bacteria and 24–32 nt long in eukaryotes, that is displaced by a DNA helicase. The resulting gap in DNA can be filled in by a DNA polymerase to leave a nick that is sealed by DNA ligase.

Conclusion

The instructions needed for producing all the components of a cell and for regulating their functions are encoded in the sequence of DNA. Accurate transmission of this information to progeny and protecting of the genome from chemical degradation are essential to life. Complementary base pairing in duplex DNA provides an elegant means for accurate replication of DNA and repair of DNA damage. Each strand of the duplex provides a template for generating the other strand and in essence acts as a "back-up copy" of the information. The many different proteins and enzymes


A. Base Excision Repair
5'
3'
DNA glycosylase
AP endonuclease
P
deoxyribose-phosphodiesterase
DNA polymerase
DNA ligase
1 nt replaced
B. Nucleotide Excision Repair
damage recognition proteins
DNA helicase
endonuclease
endonuclease
P
DNA polymerase
DNA ligase

required to physically manipulate large DNA polymers in replication, recombination, and repair, all take advantage of the complementary base pairing between strands to accomplish their tasks. These enzymes are capable of a sufficient level of accuracy to maintain genetic integrity, yet also allow a low level of mutations to generate genetic diversity ultimately allowing a population to adapt to changing conditions. Understanding how the cellular machinery functions to replicate, recombine, and repair the genome is central to understanding evolution of species and the origin of genetic diseases.

Glossary and Abbreviations

AP Endonuclease An enzyme that cleaves the DNA backbone on the 5' side of an AP site to create a 3' hydroxyl and 5' phosphate.

AP Site A baseless sugar in DNA. AP is an abbreviation for apurinic or apyrimidinic.

Base Excision Repair A pathway that repairs damage to DNA bases by excising the damaged base and replacing it with an undamaged nucleotide.

Branch Migration Movement of a Holliday junction that unwinds DNA strands ahead of the junction and pairs them with homologous strands behind the junction.

DNA Glycosylase An enzyme that catalyzes the excision of damaged bases by cleaving the C1'-N glycosylic bond between the damaged base and the sugar.

DNA Helicase An enzyme that catalyzes the unwinding of duplex DNA.

DNA Ligase An enzyme that joins two strands of DNA at a nick by ligating a 3' hydroxyl end to a 5' phosphate end.

DNA Polymerase An enzyme that catalyzes the extension of a DNA polymer by incorporating 2'-deoxynucleoside monophosphates in a template directed reaction.

Exonuclease An enzyme that catalyzes the excision or removal of nucleotides from a DNA strand.

Fidelity Refers to the accuracy of synthesis by DNA polymerases.

Holliday Junction A four-way DNA junction formed when strands from homologous duplexes crossover during recombination.

Fig. 10. *(opposite page)* Repair of DNA by excision of the damage and resynthesis of DNA. **(A)** The base excision repair pathway begins with the removal of a damaged base by a DNA glycosylase. In this scheme undamaged DNA bases are indicated by black squares and the damaged base is indicated by a light square. The C1'-N glycosylic bond between the base and the sugar is cleaved leaving a baseless sugar residue (AP site) in DNA. The DNA strand is cut 5' to the AP site creating a 3' hydroxyl on one side of the cut and a 5′phosphate ("P") on the other. Deoxyribophosphodiesterase activity is required to excise the sugar-phosphate residue to create a one nucleotide gap that can be filled in by a DNA polymerase. Repair is complete when the strands are ligated by a DNA ligase. **(B)** The nucleotide excision repair pathway removes a short segment of DNA containing a damaged base (red starburst). The damaged base is recognized and bound by a protein complex. This protein complex serves to direct the other proteins to the site of damage so that it can be repaired. A DNA helicase separates the DNA strands on either side of the damaged nucleotide. Specific endonucleases recognize the forked single-stranded/double-stranded DNA junctions at these sites and cleave the DNA at the junctions. The DNA strand is cleaved 3' to the damaged nucleotide followed by cleavage on the 5' side. The gap created by excision of the damaged DNA segment is filled in by a DNA polymerase and the two strands are joined by a DNA ligase.

Lagging Strand The strand synthesized in discontinuous segments called Okazaki fragments.

Leading Strand The strand of DNA synthesized in one continuous piece in the direction of replication fork progression.

Nick A single-stranded break in a DNA duplex.

Nucleotide Excision Repair A pathway that repairs DNA damage by removing a short segment of DNA containing the damage and resynthesizing that segment.

Okazaki fragments Discontinuous segments of DNA that are synthesized on the lagging strand.

Origin of replication A site in the genome where DNA synthesis is initiated during replication.

Primase An enzyme that synthesizes short segments of RNA to prime synthesis by DNA polymerases.

Processivity Refers to the number of nucleotides that a DNA polymerase can incorporate in a single DNA binding event.

Proofreading The process of removing incorrectly paired nucleotides catalyzed by the 3' to 5' exonuclease activity of DNA polymerases.

Replication fork The branched DNA structure formed when a DNA helicase separates the two complementary strands of DNA during replication.

Replisome The ensemble of enzymes that function at the replication fork to duplicate DNA.

Semi-Conservative Refers to replication of DNA that produces duplexes composed of one newly synthesized and one original strand.

Semi-discontinuous Refers to DNA synthesis that produces one continuous strand on the leading strand and Okazaki fragments or discontinuous strands on the lagging strand.

Strand invasion Pairing of a single-stranded DNA with a homologous region of duplex DNA to form a D-loop.

Suggested Reading

DNA Replication

Davey, M. J. and O’Donnell, M. (2000) Mechanisms of DNA replication, Curr. Opin. Chem. Biol. 4, 581–586.

Goodman, M. F. (2000) Coping with replication “train wrecks” in *Escherichia coli* using Pol V, Pol II, and RecA proteins, Trends Biochem. Sci. 25, 189–195.

Gulbis, J. M., Kelman, Z., Hurwitz, J., O’Donnell, M., and Kuriyan, J. (1996) Structure of the C-terminal region of p21WAF1/CIP1 complexed with human PCNA, Cell 87, 297–306.

Kelly, T. J. and Brown, G. W. (2000) Regulation of chromosome replication, Annu. Rev. Biochem. 69, 829–880.

Kong, X.-P., Onrust, R., O’Donnell, M., and Kuriyan, J. (1992) Three-dimensional structure of the β subunit of *E. coli* DNA polymerase III holoenzyme: A sliding DNA clamp, Cell 69, 425–437.

Kunkel, T. A. and Bebenek, K. (2000) DNA replication fidelity, Annu. Rev. Biochem. 69, 497–529.

Waga, S. and Stillman, B. (1998) The DNA replication fork in eukaryotic cells. Annu. Rev. Biochem. 67, 721–751.

DNA Recombination

Haber, J. E. (2000) Partners and pathways repairing a double-strand break, Trends Genet. 6, 259–264.

Kowalczykowski, S. C. (2000) Initiation of genetic recombination and recombination-dependent replication, Trends Biochem. Sci. 25, 156–165.

West, S. (1997) Processing of recombination intermediates by RuvABC proteins, Annu. Rev. Genet. 31, 213–244.

DNA Repair

Batty, D. P. and Wood, R. D. (2000) Damage recognition in nucleotide excision repair of DNA, Gene 241, 193–204.

de Laat, W. L., Jaspers, N. G. J., and Hoeijmakers, J. H. J. (1999) Molecular mechanism of nucleotide excision repair, Genes Dev. 13, 768–785.

Lindahl, T. (1993) Instability and decay of the primary structure of DNA. Nature 362, 709–715.

Lindahl, T. (2000) Suppression of spontaneous mutagenesis in human cells by DNA base excision-repair, Mutation Res. 462, 129–135.

5 Transcription, RNA Processing, and Translation

Thomas P. Yang and Thomas W. O'Brien

Introduction

The processes of transcription, RNA processing (in eukaryotes), and translation constitute the pathway that leads to the conversion of genetic information in the linear sequence of bases in genomic DNA into the linear amino acid sequences of functional proteins. Thus, DNA undergoes *transcription* to synthesize a primary RNA transcript, which in eukaryotes undergoes *RNA processing* to produce a mature messenger RNA (mRNA), then mRNAs are *translated* into functional polypeptides. Each of these cellular processes will be described below.

Transcription

The process of transcription involves the sequential and enzyme-catalyzed polymerization of ribonucleotide triphosphates into a single-stranded linear RNA molecule that is complementary to, and encoded by, one strand of a DNA template. This process in eukaryotes occurs in the nucleus. The growth of the nascent RNA chain proceeds from the 5' end to the 3' end of the chain, elongating by adding nucleotides to the –OH group at the 3' end of the RNA. As depicted in Fig. 1, polymerization occurs by formation of a phosphodiester bond between the –OH group of the ribose moiety at the 3' end of the elongating RNA and the 5' phosphate of the ribonucleotide triphosphate (rNTP) precursor to be added to the growing RNA chain. Thus, unidirectional growth of the RNA chain occurs in the 5' to 3' direction. The phosphodiester bond is synthesized by a condensation reaction involving the 3' –OH group of the sugar and the α phosphate group of the rNTP, with release of pyrophosphate (PPi). As shown in Fig. 1, the nucleotide sequence of the elongating RNA chain is specified by the nucleotide sequence of one strand of the duplex DNA template. Following the rules of Watson-Crick base pairing, adenine, cytosine, guanine, and thymine in the template DNA sequence direct the addition of uracil, guanine, cytosine, and adenine, respectively, to the RNA sequence. In most cases, only one strand of a given region of double-stranded DNA is transcribed into RNA, though a small but increasing number of eukaryotic genes show transcription from both strands over all or a portion of the gene. In several cases, a gene encodes a functional *sense* transcript as well as a so-called *anti-sense* transcript (often of unknown function) that extends over the entire region encoding the

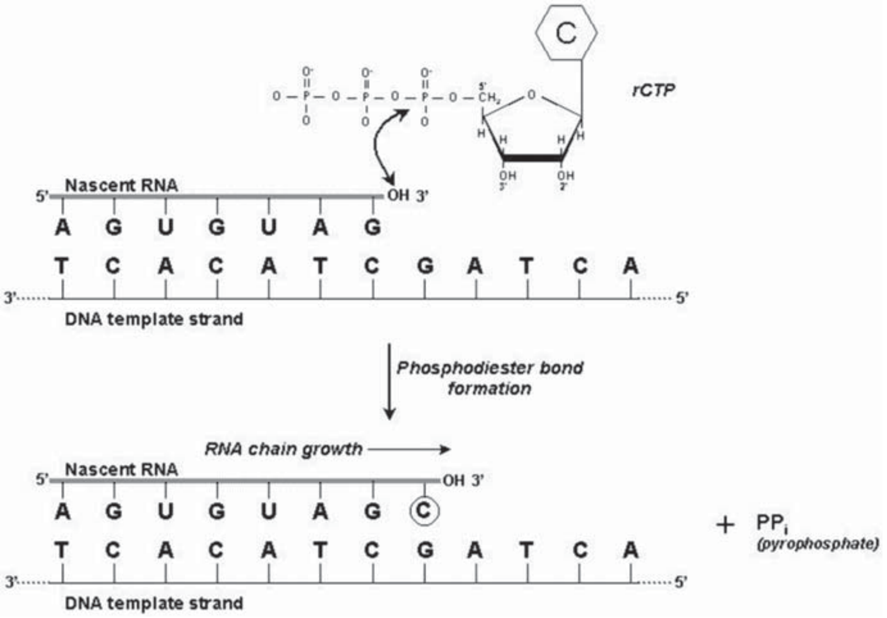


Fig. 1. DNA-dependent synthesis of RNA. In this example, a cytidine triphosphate (rCTP) precursor is added to the 3' end of an elongating RNA chain by forming a phosphodiester bond between the 5' phosphate of the rCTP precursor and the 3' OH of the previously added nucleotide. The nucleotide sequence of the nascent RNA is specified by the complementary nucleotide sequence of the DNA template strand according to Watson-Crick base pairing. The circled C residue indicates the position of the newly added nucleotide.

sense transcript. Some of these anti-sense transcripts have been postulated to regulate the expression of the sense transcript.

The synthesis of RNA from a DNA template is catalyzed by the enzyme RNA polymerase. Most, if not all, RNA synthesis in prokaryotes is directed by a single RNA polymerase, while RNA synthesis in eukaryotes is catalyzed by three different RNA polymerases, pol I, pol II, and pol III, each of which transcribes a different class of genes. The high-resolution structure of both bacterial RNA polymerase and eukaryotic RNA polymerase II has recently been determined by X-ray crystallography. The amino acid homology of certain subunits in the prokaryotic and eukaryotic polymerases and their many shared structural features are notable though not unexpected due to their similar functions.

The macromolecular complex formed during the process of transcription consists of RNA polymerase, the DNA template, and the nascent RNA (*see* Fig. 2). The double-stranded DNA template is melted to form a transcription *bubble* (of approx 12 base pairs) within the bubble the elongating nascent RNA forms an approx 9 bp RNA:DNA duplex with the template strand of the DNA at its 3' end. As the polymerase moves downstream along the DNA template, the double-stranded DNA helix is unwound at the front of the bubble and rewound behind the bubble. Growth of the nascent RNA occurs by the addition of nucleotides at the 3' end of the nascent RNA within the bubble, with nucleotide triphosphate precursors presumably translocated to the active site of the enzyme via a pore and channel within the structure of the polymerase.

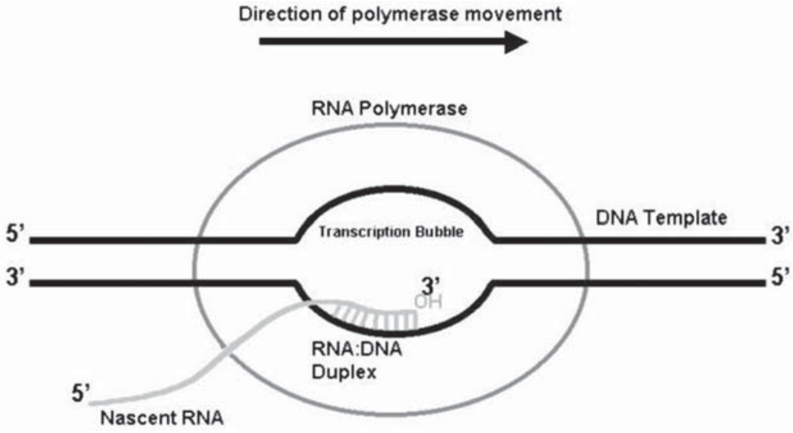


Fig. 2. Schematic of the transcription elongation complex.

Regulation of transcription, particularly regulation of transcription initiation, is a major mechanism that regulates macromolecular biosynthesis in the cell. However, the processes by which transcription is initiated and regulated in prokaryotes and eukaryotes are notably different. Therefore, a description of each of these processes in prokaryotes and eukaryotes is presented.

Transcription in Prokaryotes

Transcription of prokaryotic genes is accomplished by a single RNA polymerase. The prototype of the prokaryotic RNA polymerase is the polymerase of *Escherichia coli*. It is composed of four different polypeptide subunits, α, β, β', and σ. The RNA polymerase core enzyme, which functions in transcription elongation, is composed of two α subunits, and one each of the β and β' subunits. The RNA polymerase holoenzyme, which initiates transcription, is organized as the core enzyme plus one copy of the σ subunit.

To initiate transcription, RNA polymerase holoenzyme first binds to DNA sequences immediately upstream of a gene. This region upstream of the gene, termed *the promoter*, contains DNA sequences specifically recognized and bound by the RNA polymerase holoenzyme. The σ subunit confers upon the holoenzyme the ability to recognize and bind to the promoter in a DNA sequence-specific manner, and to initiate transcription at a site specifically from the template strand of the DNA. Different σ subunits, such as σ^{54} and σ^{70}, recognize and bind different subsets of prokaryotic promoter sequences. For example, σ^{70}-containing holoenzyme recognizes and contacts (via the σ subunit) specific promoter sequences surrounding positions –10 and –35 (upstream of the transcription initiation site).

Initial binding of the polymerase to the promoter leads to formation of a *closed* binary complex where the polymerase-bound DNA duplex remains double-stranded. This closed complex is then converted to a more stable *open* complex where melting of the DNA duplex occurs and the two strands of the promoter DNA undergo separation. The DNA duplex is melted over ~12 bp, from the –10 region to just downstream of the transcription initiation site (i.e., the deoxynucleotide in the DNA, encoding the first ribonucleotide in the RNA transcript). Both the closed and open promoter complexes are composed of multiple intermediate forms, with formation of an open complex accompanied by a major conformation change in the RNA polymerase. Conversion to an open promoter complex is followed by entry of ribonucleotide triphosphates (rNTPs) and binding of the initial (rNTP) to form a ternary complex. For σ^{70} promoters, conversion and stability of the open promoter complex does not require rNTP hydrolysis, though for some promoters (e.g., certain σ^{54} promoters) this conversion from a closed to open complex requires interaction with activator proteins bound to the DNA upstream of the promoter as well as hydrolysis of ATP. After forming the open ternary complex, active transcription begins with the synthesis of a series of abortive transcripts. These are short transcripts <10 nucleotides in length that are repetitively synthesized and released without release of the polymerase from the promoter. Once a nascent transcript reaches a length of 8–10 nucleotides, σ factor is released from the holoenzyme and RNA polymerase escapes and clears the promoter to begin committed elongation of the transcript. The transcription elongation complex (TEC) is a stable association of the core polymerase with the DNA template and nascent RNA. Elongation proceeds by movement of the TEC down the linear DNA template, with sequential and continuous addition of nucleotides to the 3' end of the nascent transcript. The exact ribonucleotide added to the elongating RNA chain is specified by the next base in the DNA template sequence according to Watson-Crick base pairing (i.e., A:U, C:G, G:C, T:A for DNA:RNA base pairs). Misincorporation of a nucleotide in the nascent RNA, or pausing of the polymerase along the DNA template during elongation, leads the polymerase to backtrack along a short stretch of the DNA, cleavage of the newly synthesized portion of the nascent RNA at the 3' end, and continuation of elongation from the newly truncated nascent RNA (including resynthesis of the cleaved portion of the RNA).

Termination of transcription in prokaryotes occurs at specific sites downstream of the coding region of genes and is accomplished by either of two possible mechanisms. One, termed *rho-dependent termination*, is mediated by the action of the rho termination protein and requires the hydrolysis of ATP. The second mechanism is rho-independent termination and occurs via formation of specific hairpin structures in the nascent RNA that destabilizes the ternary complex immediately downstream of the hairpin and leads to dissociation of the ternary complex and transcription termination. Rho-independent termination can also occur via terminators formed by an RNA:DNA hybrid between the nascent RNA and the DNA template just upstream of the elongation complex.

Following termination of transcription and release of the RNA polymerase from the DNA template and the RNA transcript, the free core polymerase is able to rebind another σ factor and re-initiate another round of transcription.

Regulation of transcription in prokaryotes occurs by a wealth of different mechanisms that often involve activator and repressor proteins. Descriptions of the diverse mechanisms of prokaryotic transcriptional regulation are beyond the scope of this chapter.

Initiation and Regulation of Transcription in Eukaryotes

The three eukaryotic RNA polymerases all synthesize RNA transcripts encoded by a DNA template, but transcribe different subsets of genes. RNA pol I synthesizes the precursor of ribosomal RNAs (rRNAs) which is eventually processed into the mature 5.8S, 18S, and 28S rRNAs. The bulk of RNA synthesis in a cell is carried out by pol I. RNA pol II primarily transcribes the mRNA encoding genes, and therefore is responsible for transcription of the largest and most diverse subset of genes in the cell. RNA pol III catalyzes synthesis of an assortment of small RNAs including tRNAs and 5S rRNA. Unlike the prokaryotic RNA polymerase, highly purified eukaryotic RNA polymerases do not recognize promoter sequences by themselves or initiate transcription by binding directly to specific DNA sequences in eukaryotic promoters. Rather eukaryotic polymerases are recruited to promoter regions via protein-protein interactions by both proteins that bind to specific DNA sequences in promoter and other regulatory regions as well as additional proteins that interact with these DNA-binding proteins.

The following description of eukaryotic transcription and regulation will focus on mechanisms of pol II transcription because of the diversity of genes it transcribes and the current intensive efforts to elucidate mechanisms of pol II transcription and regulation.

RNA Polymerase II and Pol II Core Promoters

Eukaryotic RNA polymerase II (pol II) consists of 12 subunits totaling greater than 500 kD in size, with all subunits very similar between yeast and humans. Most human subunits can function in place of their yeast homolgues, and 9 pol II subunits are highly conserved among the three different eukaryotic RNA polymerases. Furthermore, the two largest pol II subunits, Rpb1 and Rpb2, show significant similarity to the β' and β subunits of bacterial RNA polymerase, respectively, while two other pol II subunits show lesser similarity to the α subunit of the bacterial enzyme. The structure of yeast pol II has recently been determined by X-ray crystallography at 2.8 angstrom resolution. The largest pol II subunit, Rpb1, contains a conserved tandem repeat heptapeptide sequence at the C-terminus termed the *C-terminal domain* (CTD). The CTD and its phosphorylation state play a critical role in both transcription and RNA processing. An unphosphorylated CTD is associated with the pol II initiation complex, whereas a phosphorylated CTD is correlated with the elongation phase of transcription and association of the pol II complex with RNA processing factors.

Promoters transcribed by pol II exhibit a diverse structure. Most consist of a core promoter (sometimes referred to as the basal promoter) located in the immediate vicinity of the transcription initiation site, as well as nearby regulatory sequences that activate or repress activity of the core promoter, usually in response to e.g., cellular conditions, extracellular signals and environment, cell type, and stage of development. In addition, the function of many eukaryotic promoters, particularly in higher eukaryotes, is positively and/or negatively modulated by regulatory DNA sequences that can be located within the gene, upstream of the gene and/or downstream of the gene at distances exceeding 50 kb of DNA.

The core promoter contains cis-acting elements (i.e., DNA sequences) that can act as a nucleation site for the formation of the pre-initiation complex (PIC) and recruitment of pol II. Core promoter elements in association with the PIC also determine the site(s) where transcription of each gene is initiated. These core elements frequently include the so-called TATA box; however, other core DNA elements can either replace the function of the TATA box in TATA-less promoters, or can act in conjunction with the TATA box to facilitate transcription initiation. These other cis-acting elements in the basal promoter include initiator (Inr) elements, a downstream promoter element (DPE), and a TFIIB recognition element (BRE). These cis-acting elements function by interacting with sequence-specific DNA-binding proteins that initiate and/or mediate PIC assembly.

Formation of the Pre-Initiation Complex

A multi-protein complex, termed the *pre-initiation complex* (PIC) and consisting of a series of general transcription factors (GTFs) as well as pol II, is assembled at the core promoter as the first step of transcription initiation. These GTFs include the transcription factors TFIIA, TFIIB, TFIID, TFIIE, TFIIF, and TFIIH, though not all promoters require all of these factors. In vitro studies suggest that the PIC is assembled at core promoters in a sequential fashion (*see* Fig. 3). For genes with a canonical TATA box in the core promoter, this sequential assembly is initiated by the sequence recognition and binding of the multi-subunit GTF TFIID to the TATA box; TFIID consists of a DNA-binding subunit termed the *TATA box-binding protein* (TBP) that recognizes and binds the TATA sequence, plus a variety of TBP-associated factors (TAFs). TBP appears to be a common component of the PIC formed at pol I, pol II, and pol III promoters. Binding of TFIID to the core promoter DNA of pol II genes is followed by binding of TFIIA and TFIIB to the TFIID-TATA box complex, then recruitment of a pre-formed complex of pol II, TFIIF and TFIIE, and finally binding of TFIIH. TFIIA appears to stabilize the binding of TBP to DNA, and TFIIB may function in selecting the position of the transcription initiation site. TFIIF bound to pol II suppresses non-specific binding of the polymerase to DNA and stabilizes the PIC. TFIIE may be involved in recruiting TFIIH and melting the DNA duplex when forming the open promoter complex. The multi-subunit complex that comprises TFIIH contains an ATP-dependent helicase activity, a DNA-dependent ATPase, and CTD kinase activity. Thus, TFIIH appears to be involved in forming the open promoter complex via its helicase activity, and also may be required for the conversion of abortive transcription to committed elongation. The kinase activity of TFIIH phosphorylates the CTD of pol II and, therefore, may be involved in triggering the functions associated with a phosphorylated CTD. In addition, the observed coupling of transcription and DNA repair appears to be due to the common subunits shared by TFIIH in the transcription complex and the nucleotide excision repair complex.

However, a significant percentage of eukaryotic promoters lack a canonical TATA box and/or contain other core promoter elements such as an Inr (initiator element), DPE (downstream promoter element), and BRE (TFIIB recognition element). This leads to a diversity of mechanisms for forming the PIC at different promoters and for the activation of basal transcription from the core promoter. TATA-less promoters often contain an Inr element that serves as the nucleation site for assembly of PIC. The Inr is a short conserved DNA sequence spanning the transcription initiation site that acts as the binding site for several different DNA-binding initiator proteins such as YY1, TFIII, USF, $TAF_{II}250$, and $TAF_{II}150$. The Inr (bound by its transcription factors) can function alone in TATA-less promoters, or in combination with a TATA box or DPE (if these elements are present) to nucleate assembly of the PIC at core promoters. If present in a core promoter, the DPE is typically located approx 30 bp downstream of the transcription initiation site and is recognized and bound by subunits of the general transcription factor TFIID. The BRE element is located immediately upstream of the TATA box in certain core promoters with TATA boxes and facilitates the interaction and binding of TFIIB to the core promoter. In addition, a

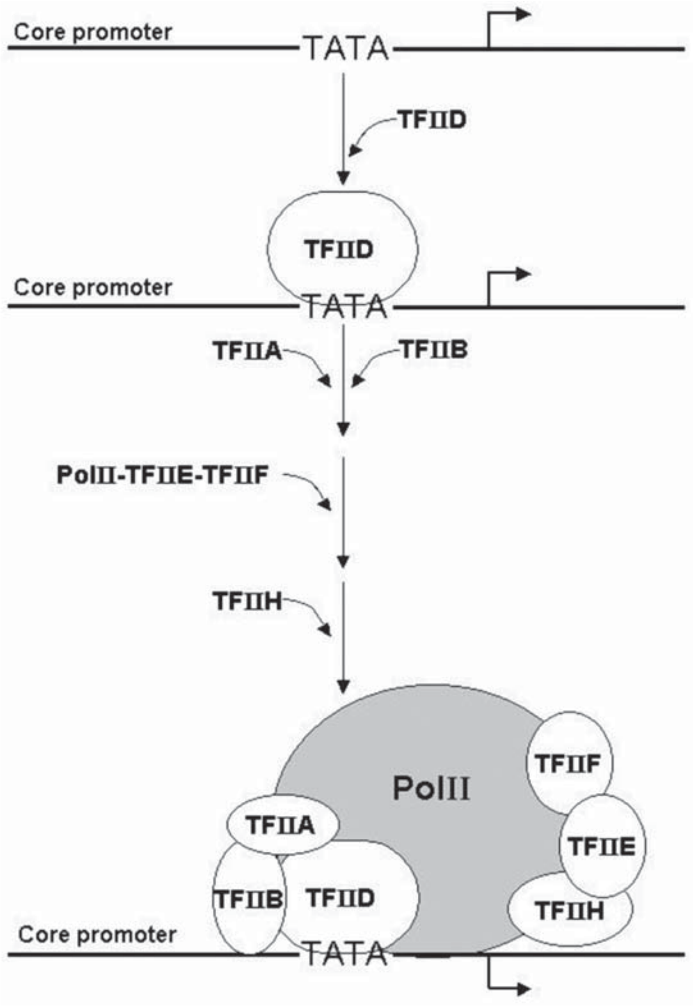


Fig. 3. Model for sequential assembly of the general transcription factors (GTFs) and pol II into the pre-initiation complex (PIC). The right angled arrow indicates the position and direction of transcription initiation.

subset of promoters, commonly those that lack TATA boxes, do not initiate transcription at a discrete nucleotide, but initiate transcription at multiple sites within the core promoter.

Overall, assembly of the PIC on the various forms of the core promoter serves to recruit and stabilize binding of pol II to the promoter, positioning pol II at the correct site on the gene for initiating transcription. It helps form an open transcription initiation complex by melting of the DNA duplex, and in the conversion of the PIC from transcription initiation to the transcription elongation complex. The open transcription initiation complex formed at the core promoter is associated with a melted DNA duplex of 12–15 bp that extends halfway from the transcription initiation site to the TATA box.

Activators, Co-Activators, Enhancers

Transcription from the core promoter is usually augmented by the action of additional transcriptional activator proteins that bind regulatory DNA sequences outside (upstream and/or downstream) of the core promoter. These activator (or repressor) elements serve to increase (or decrease) levels of transcription above the basal activity of the core promoter. They often act in the context of cell type, e.g., cellular conditions, stage of development, and response to stimuli such as hormones and growth factors. The combinatorial action of multiple activators and repressors on a given gene under a given set of cellular conditions determines the transcriptional status of the gene and governs its level of expression.

Activator proteins bound to their cis-acting regulatory sites appear to function by at least two mechanisms. Some activators bind their cognate site in the vicinity of the core promoter and appear to facilitate recruitment of certain GTFs to the core promoter and assist in formation of the PIC. These activators are presumed to assist PIC formation via protein-protein interactions between the activator and its target component(s) of the PIC. Alternatively, some activators may function by stimulating activity of the PIC and facilitating release of the PIC from the core promoter and/or facilitating transcription elongation. Some activators also appear to act by participating in the recruitment of chromatin modifying and remodeling complexes. In many cases of activator function, contact between the DNA-bound activator protein and the PIC requires bending or looping of the intervening DNA. Thus, some classes of activator proteins function by facilitating the bending of the DNA between the bound activator and the core promoter.

However, not all DNA-binding activator proteins that act on assembly or functions of the PIC can operate by direct contact with components of the PIC. Some activators require the action of an intermediary protein or protein complex between the activator and the PIC. These co-called co-activators act through protein-protein interactions as a bridge between the DNA-bound activator protein and the PIC at the basal promoter. Thus, the function of many activator proteins are carried out via association with co-activator proteins or protein complexes. A protein complex (of ~20 subunits in yeast) that appears to play an important role in activated transcription is the *mediator* complex. It functions as a general co-activator of transcription and appears to mediate the action of a variety of activator proteins. The mediator complex is bound to the unphosphorylated form of the CTD of pol II and has been found to be a component of the pol II holoenzyme complex.

In addition to cis-acting activator sequences that bind activator proteins in the vicinity of the core promoter, other cis-acting activator sequences are found at larger distances from the core promoter (up to tens of thousands of kb) both upstream, downstream and within genes. Some of these distal activator sequences are thought to function by looping the intervening DNA to contact the transcriptional machinery in the vicinity of the core promoter. A subset of these long-range activators, termed *enhancers*, have been found to act regardless of their distance from the promoter, location relative to the promoter (i.e., upstream and downstream) and either in the forward or reverse orientation of the cis-acting activator element. These enhancer sequences have been shown to contain a variety of binding sites for a constellation of DNA-binding transcription factors that are presumed to form large DNA-protein complexes that activate transcription.

A large enhancer-like region spanning several thousand bps in mammals, termed a *locus control region* (LCR), also activates transcription at a distance, and can regulate activation of multiple co-regulated genes within a gene cluster such as the β-globin gene family. The characteristic feature of LCRs is their ability to allow ectopically integrated transgenes to be regulated independent of the effects from the flanking DNA and chromatin at their new integration sites. LCRs also have been postulated to be involved in opening of chromatin structure across entire loci or domains; however, this role for LCRs is currently unclear.

Pol II Holoenzyme

Recently, holoenzyme complexes containing pol II have been purified intact from eukaryotic cells. These heterogeneous pol II holoenzyme complexes include pol II, a subset of the GTFs and the mediator complex which is believed to be bound to the CTD of pol II. This finding of a pol II holoenzyme complex suggests that formation of the PIC in vivo may not always involve step-wise and sequential assembly of the GTFs and pol II and that much of the PIC, as well as co-activators required for activation of the basal promoter, may be preassembled in vivo and brought to the core promoter intact. Nonetheless, the lack of TBP in these pol II holoenzyme complexes suggest that, in part, formation of the PIC with these holoenzyme complexes may still be initiated by TBP binding to the TATA box followed by recruitment of the holoenzyme.

The Role of Chromatin Structure

A major mode of transcriptional regulation in eukaryotes involves altering the chromatin structure of genes and regulatory regions to repress and/or activate transcription. DNA is packaged with basic histone proteins within the eukaryotic nucleus to form chromatin. The basic unit of packaging genomic DNA into chromatin is the nucleosome, a discrete nucleoprotein complex composed of ~200 bp DNA wrapped around a highly structured core of the histone proteins H2A, H2B, H3, and H4. Linear arrays of nucleosomes resembling "beads on a string" form the first level of packaging DNA in the nucleus; organization of the linear nucleosomal array into higher-order configurations continue the packaging of genomic DNA into the chromatin found in eukarotic nuclei. The structures of these higher-order organizations of nucleosomal arrays are still not well-defined.

The formation of nucleosomes over promoters and other regulatory regions generally has a repressive effect on the transcription of genes by blocking or otherwise inhibiting the binding of specific transcription factors to their cognate regulatory DNA sequences. Furthermore, in higher eukaryotes, the higher-order packing of nucleosomal arrays also appears to regulate transcription by governing the general access of an entire locus (with its regulatory regions) to components of the transcriptional machinery. Thus, modulating the higher-order chromatin structure of an entire gene or *domain*, as well as altering the position and structure of individual nucleosomes over regulatory DNA sequences, constitute critical mechanisms regulating transcription of eukaryotic genes. A number of transcriptional activators (and repressors) appear to act by recruiting and regulating the function of chromatin modifying and remodeling complexes within specific genes.

The position and structure of individual nucleosomes associated with critical regulatory regions of genes has recently been shown to be regulated by ATP-dependent, multi-subunit complexes that remodel or modify nucleosomes or localized nucleosomal arrays (e.g., Swi/Snf, NURD, CHRAC, RSC). Some of these nucleosome remodeling complexes appear to act by repositioning (e.g., sliding, and/or transferring) intact nucleosomes from regulatory regions or sites, and presumably *uncover* previously obstructed binding sites in DNA for sequence-specific DNA-binding transcription factors. Furthermore, a number of remodeling complexes also appear capable of subtly altering the structure of individual intact nucleosomes by changing the interactions of the DNA with the histone core. This allows transcription factors access to their otherwise obstructed DNA binding sites within native nucleosomes.

Nucleosome structure is also altered by the activity of various nucleosome modifying complexes. These complexes covalently modify the side chains of amino acids (particularly lysines) in the N-terminal region of histones, thereby altering the structure of the associated nucleosomes and/or nucleosomal arrays. Examples of these nucleosome modifying complexes include histone acetylation and deacetylation complexes, as well as histone methylases. Acetylation of histone H4 has been associated with actively transcribed chromatin, while underacetylation of histone H4 has been associated with transcriptionally repressed chromatin. Presumably, these histone modifiying activities alter chromatin structure and either facilitate or preclude the access of transcription factors to their sites of action. Regulation of higher chromatin structure (i.e., the organization of nucleosomal arrays into higher-order structure) across entire genes or domains is not well-understood, and mechanisms and regulatory elements that carry out these structural alterations of chromatin have not been definitively identified. This is further discussed in Chapter 6.

In summary, the opening of higher-order chromatin structure, the modification and/or positioning of nucleosomes to allow transcription factors to bind their cognate DNA binding sites, and the action of activators, enhancers, etc. all facilitate formation of a stable and functional PIC. Assembly of the transcriptional machinery at the promoter is associated with formation of a stable open promoter complex, leading to clearance of the pol II complex from the promoter and elongation of the nascent RNA.

Elongation

As with prokaryotic transcription elongation, the initial phase of RNA chain elongation is associated with abortive transcription, the repetitive synthesis of very short RNA molecules that are synthesized, released, and re-initiated. After the polymerization of four or more nucleotides into a nascent RNA, the transcript becomes stabilized in the complex. In part, this permits a transition from transcription initiation to promoter release (or clearance) and productive elongation. Conversion of the pol II complex from transcription initiation to elongation is accompanied by phosphorylation of the CTD of pol II, release of certain general transcription factors GTFs and the mediator complex, and association of TFIIS and other elongation factors as well as certain RNA processing factors. However, details of the process that converts the initation complex to transcription elongation are not well-understood.

Specific factors have been identified that facilitate promoter clearance (e.g., P-TEFb) while other factors act negatively to repress promoter clearance (e.g., NELF, DSIF). Positive elongation factors that facilitate promoter clearance also include the GTFs TFIIF and TFIIH. Other factors facilitate the processivity of elongation by the pol II complex such as TFIIS, elongin, and CSB. The elongation complex appears to contain 9 bp of RNA:DNA hybrid within the transcription bubble, with the front of the bubble positioned 3 nucleotides before the beginning of the RNA:DNA hybrid. As with prokaryotic transcription, the pol II elongation complex catalyzes the sequential addition of ribonucleotides to the 3' end of the elongating mRNA. The exact linear sequence of ribonucleotides in the mRNA is dictated by the nucleotide sequence of the template DNA strand based upon Watson-Crick base pairing.

TERMINATION

Termination of eukaryotic gene transcription differs significantly from termination in prokaryotes. Rather than terminating transcription at discrete sites as seen in prokaryotes, termination in eukaryotes does not appear to occur at specific sites, but appears to occur by loss of processivity or stability of the elongation complex downstream of the gene. Thus, 3' end formation of eukaryotic mRNAs does not occur as a direct result of transcription termination at a specific site, but occurs by post-transcriptional processing of the RNA transcript where sequence-specific cleavage of the transcript at a specific site in the RNA is followed by post-transcriptional polyadenylation of the 3' end of the cleaved transcript. Evidence suggests that the polyadenylation signal and/or the site-specific cleavage of the nascent RNA may be involved in signaling or facilitating transcription termination downstream.

RNA Processing

Generating a mature eukaryotic mRNA molecule from the primary transcript produced by transcription requires a series of post-transcriptional RNA processing steps that include 5' mRNA capping, 3' polyadenylation, RNA splicing, and RNA editing. Alternative sites for polyadenylation and RNA splicing within the same gene also play a significant role in post-transcriptional regulation of gene expression. As discussed in the following, several of these processes appear to be directly coupled to transcription via association with the elongating pol II complex.

5' Capping

After the nascent RNA transcript has reached the ~25–30 nucleotides, the 5' end of the transcript is covalently modified by enzymatic addition of a guanine triphosphate moiety via an unusual 5'–5' covalent linkage (*see* Fig. 4A). This novel bond structure between the first and second nucleotides at the 5' end of the mature mRNA protects the mRNA from degradation by a 5'→ 3' exonuclease activity in the cell. Formation of this 5' cap structure is catalyzed, in part, by guanylyltransferase, an enzyme that is bound to the phosphorylated form of the CTD of pol II. Subsequent to the capping reaction, methytransferases add methyl groups to the N7 position of the newly added guanine to generate the 7-methylguanosine cap, and a methyl group to the 2' hydroxl group of the first and second (in vertebrates) ribose moieties of the nascent transcript.

Polyadenylation

Formation of the 3' end of eukaryotic mRNAs is accomplished by enzymatic cleavage of the nascent RNA transcript followed by the post-transcriptional enzymatic addition of 100–250 adenosine residues to the 3' end of the mRNA (*see* Fig. 4B). Cleavage and polyadenylation is initiated by recognition of the polyadenylation signal sequence AAUAAA in the RNA transcript, followed by cleavage of the transcript at a discrete site 10–25 nucleotides downstream of the sequence, and polyadenylation of

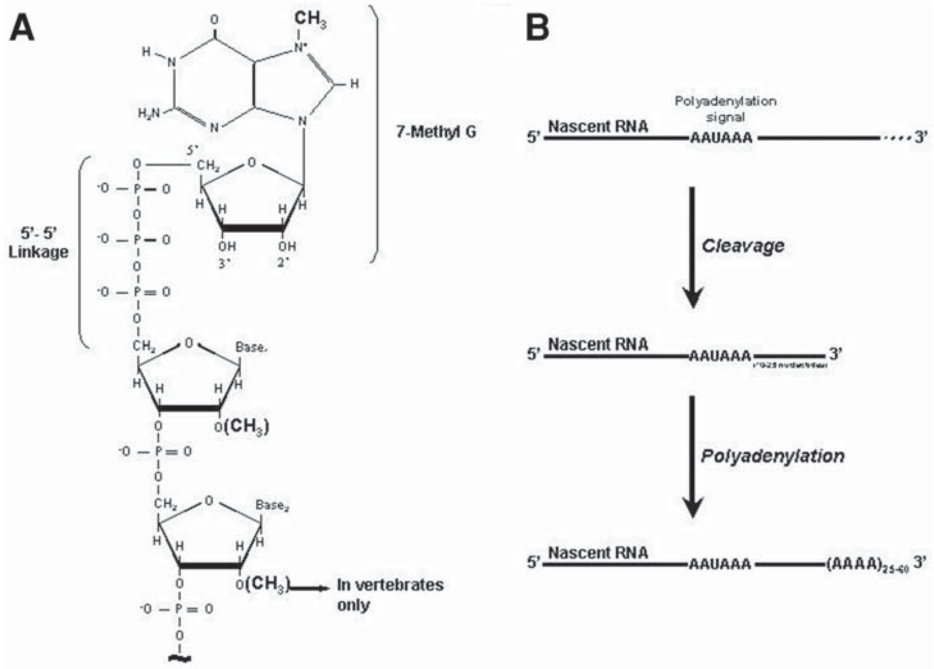


Fig. 4. Capping and polyadenylation. **(A)** The 5' mRNA cap structure. **(B)** 3' polyadenylation of mRNA.

the 3' end of the upstream fragment generated by the cleavage reaction. The downstream RNA fragment generated by transcript cleavage is rapidly degraded. Transcript cleavage and polyadenylation is accomplished by a well-characterized, multi-subunit complex that assembles over the polyadenylation signal sequence of the RNA transcript. Addition of A residues to the 3' end of the cleaved transcript is performed by poly(A) polymerase, a component of the complex. In addition to the apparent coupling of transcription termination with polyadenylation, polyadenylation is also coupled to transcription elongation by binding of polyadenylation factors (e.g., the cleavage stimulatory factor CStF) to the CTD of pol II. In fact, some factors involved in polyadenylation appear to associate with the CTD at the time of transcription initiation at the promoter. As a mechanism of post-transcriptional gene regulation, alternative sites of polyadenylation within the same gene can generate different protein products, e.g., in different cell types, different stages of development, or in response to extracellular signals.

RNA Splicing

Many eukaryotic genes, particularly in higher eukaryotes, contain DNA sequences that interrupt the nucleotide sequence that encodes the mRNA produced from the gene. These intervening sequences, termed *introns*, are transcribed and included in the primary RNA transcript generated by the process of transcription. These introns are then precisely removed post-transcriptionally and the coding portions of the mRNA, termed *exons*, are rejoined by the process of RNA splicing. A single gene can contain multiple introns, some extending for tens of thousands of bases between exons. RNA splicing is carried out in the nucleus on the nascent RNA by a large ribonucleoprotein complex termed the *spliceosome*. The spliceosome, in turn, is composed of proteins and a variety of small nuclear ribonucleoprotein particles (snRNP's), each containing a small nuclear RNA (and associated proteins) that participate in the splicing reaction. The sequence of reactions leading to the removal of an intron and covalent joining of two exons is carried out during the sequential assembly and disassembly of the spliceosome.

Nucleotide sequences in the nascent primary RNA transcript are recognized by components of the spliceosome and mediate the splicing reaction. As shown in Fig. 5A, these sites in the nascent RNA include: 1) the GU and AG sequences found at the 5' and 3' ends, respectively, of most introns; 2) the A residue located in the intron (20–50 nucleotides upstream of the 3' end of the intron) that will participate in formation of the lariat intermediate; 3) a pyrimidine-rich region just upstream of the 3' splice site; and 4) the splicing enhancer sequences in the RNA transcript that facilitate the splicing reaction.

As depicted in Fig. 5B, the removal of an intron between two coding region exons is initiated by a transesterification reaction that breaks the ester bond between the last nucleotide of exon 1 (i.e., the upstream exon) and the first nucleotide at the 5' end of the intron, and forms a new ester bond between the 2' oxygen of the branchpoint A residue and the 5' end of the intron. This reaction forms the lariat structure characteristic of the splicing reaction. Then a second transesterification reaction breaks the phosphodiester bond between the last nucleotide of the intron and the first nucleotide of exon 2 (the downstream exon) and forms a new phosphodiester linkage between the 3' end of exon 1 and the 5' end of exon 2. This results in covalent joining of the two exons into a contiguous uninterrupted sequence in the mature mRNA and release of the intron-containing lariat structure which is then degraded.

The process of RNA splicing is highly precise; any deviation from single-nucleotide accuracy would introduce mutations into the final mRNA leading to the synthesis of defective proteins. The accuracy of the splicing reactions is mediated in part by the SR proteins, a family of structurally related serine- and arginine-rich RNA-binding proteins. The SR proteins are required for accurate recognition of exons and splice sites during assembly of the spliceosome and during the splicing reactions. Specific sequences in the nascent RNA, termed *splicing enhancers*, also are involved in *exon definition* (i.e., recognition of exons by the spliceosome) and recognition of splice sites, as well as facilitating the splicing reactions. These splicing enhancers do not appear to share an identifiable consensus nucleotide sequence, but are recognized and bound by SR proteins.

Like capping and polyadenylation, several lines of evidence indicate that RNA splicing also appears to be coupled to transcription. This evidence includes the co-localization of splicing factors in the nucleus to discrete sites of active transcription and the association of SR-like proteins to the CTD of pol II.

Alternative pathways of splicing exons within the same gene can lead to the formation of different mature mRNAs from the same gene, each of which carries a different subset of exons from the same primary transcript (*see* Fig. 5C) Thus, from the same nascent transcript, some spliced mRNAs can skip specific exons that are present in other mRNAs from the same primary transcript, or have additional exons that are absent in other mRNAs. This alternative splicing of exons from a single primary RNA transcript is highly regulated (mediated by SR proteins) and often

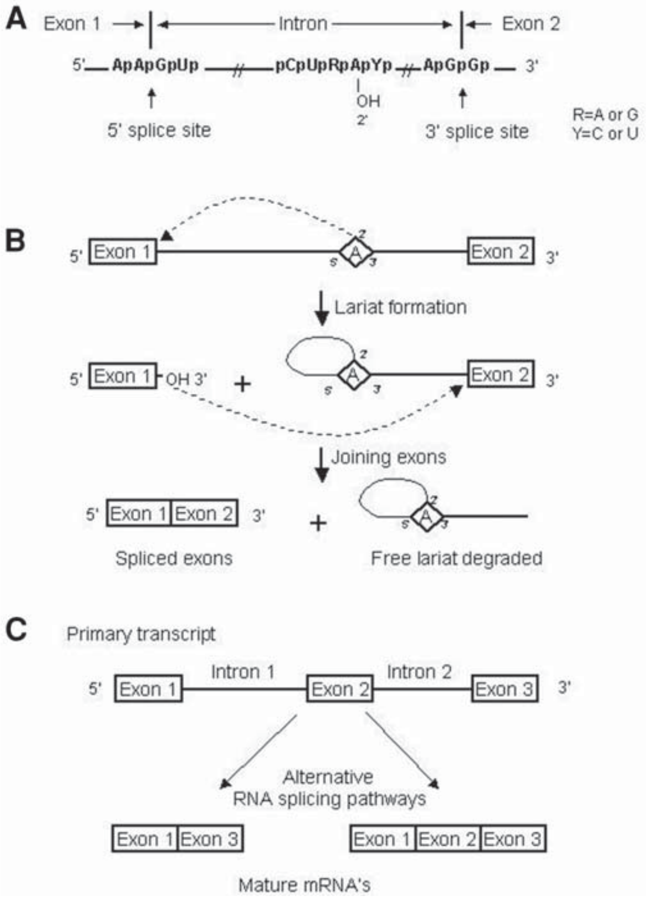


Fig. 5. Schematic of RNA splicing. **(A)** Location of conserved elements involved in RNA splicing. **(B)** Sequence of reactions in RNA splicing. **(C)** Alternative RNA splicing; a gene with three transcribed exons produces two different mRNAs via alternative splicing pathways.

leads to synthesis of different proteins with different functions from the same gene. This process constitutes a major mechanism of post-transcriptional regulation of gene expression in mammals.

RNA Editing

The final post-transcriptional processing of the RNA transcript in eukaryotes is the alteration of the transcribed nucleotide sequence in the RNA molecule to a different nucleotide sequence independent of the DNA template and transcription, thus potentially changing the amino acid coding of the mature mRNA. This RNA editing process involves changing the sequence of nucleotides in the newly synthesized RNA to a new nucleotide sequence that is not encoded in the DNA template from which the RNA was originally transcribed. This post-transcriptional modification has been identified for a relatively small number of genes in species as diverse as trypanosomes and humans. These changes in the RNA sequence can occur by inserting or deleting specific nucleotides at specific positions in the RNA transcript, or by modification of specific bases at specific sites in the RNA nucleotide sequence, thereby converting one base to another (e.g., C → U via deamination of C). RNA editing is carried out by macromolecular complexes termed the *editosome*.

Transcription of genomic DNA and the subsequent post-transcriptional processing of the primary RNA transcript generate a mature mRNA whose linear sequence of nucleotides encodes a linear sequence of amino acids. The synthesis of a functional polypeptide from a mRNA template is accomplished by the process of translation.

Translation

Overview of Protein Synthesis

The ordered interaction of well over 100 different macromolecules is required to make a single protein. The entire process takes place in a huge enzymatic machine, the *ribosome*, which in all life forms consists of two ribonucleoprotein subunits (subribosomal particles). The ribosome provides a large, dynamic platform for the sequential polymerization of amino acids according to the sequence of triplet codons in a bound messenger RNA (mRNA) molecule. Ribosomes are necessarily large, in excess of 2.5 MDa, because their substrates are large. The enzymatic substrates for protein synthesis are amino acyl-tRNAs, activated forms of the amino acids carried in ester linkage on the 3' terminal nucleotide of the various tRNAs (transfer RNAs). Each amino acyl-tRNA synthetase enzyme is specific for one of the 20 different amino acids and for all of the tRNAs having anticodons corresponding to the particular amino acid. The amino acyl-tRNAs are escorted to the ribosome as ternary complexes with initiation (IF-2) or elongation factors (EF-Tu) and GTP. The selection of particular amino acyl-tRNAs on the ribosome is made on the basis of tRNA anticodon base-pairing interactions with triplet codons that are exposed in a limited region of mRNA within the decoding center on the small ribosomal subunit. The peptidyl transferase activity of the large subunit catalyzes peptide bond formation between amino acids carried by amino acyl-tRNAs (or peptidyl-tRNAs) that are juxtaposed at specific sites (A, *amino acyl*; and P, *peptidyl*) in the intersubunit space. After each peptidyl transferase reaction, the peptidyl-tRNA and the *uncharged* tRNA which has donated an amino acid to the growing peptide chain are moved (translocated) with the aid of elongation factor EF-G, along with the mRNA, to maintain the reading frame and present the next codon in the A site of the decoding center. Next an amino acyl-tRNA having an anticodon complementary to the A site codon enters the A site and this process is repeated until a termination codon is encountered in the decoding center. Then, with the aid of protein termination factors, the protein is released from peptidyl tRNA, and another protein, RRF, *ribosome-recycling factor*, promotes disassembly of the translation complex.

Given the wealth of information about protein synthesis in bacterial systems and the recent advances in bacterial ribosome structure, it is appropriate to review protein synthesis in prokaryotes.

Elements of Translation in Prokaryotes

The Genetic Code

The 20 natural amino acids are each specified by one or more triplet codons in the mRNA. Four bases in mRNA, taken three at a time, results in 64 different triplet com-

	U	C	A	G
U	UUU Phe	UCU Ser	UAU Tyr	UGU Cys
	UUC Phe	UCC Ser	UAC Tyr	UGC Cys
	UUA Leu	UCA Ser	UAA **End**	UGA **End**
	UUG Leu	UCG Ser	UAG **End**	UGG Trp
C	CUU Leu	CCU Pro	CAU His	CGU Arg
	CUC Leu	CCC Pro	CAC His	CGC Arg
	CUA Leu	CCA Pro	CAA Gln	CGA Arg
	CUG Leu	CCG Pro	CAG Gln	CGG Arg
A	AUU Ile	ACU Thr	AAU Asn	AGU Ser
	AUC Ile	ACC Thr	AAC Asn	AGC Ser
	AUA Ile	ACA Thr	AAA Lys	AGA Arg
	AUG Met	ACG Thr	AAG Lys	AGG Arg
G	GUU Val	GCU Ala	GAU Asp	GGU Gly
	GUC Val	GCC Ala	GAC Asp	GGC Gly
	GUA Val	GCA Ala	GAA Glu	GGA Gly
	GUG Val	GCG Ala	GAG Glu	GGG Gly

Fig. 6. The genetic code, as used by most organisms. The first base of the codon (5' end) is shown in the first column, and the second base is shown in the top row. AUG (and sometimes GUG and UUG) usually serves as the initiation codon, when it occurs at the beginning of the open reading frame. The three termination codons (end) are recognized by termination factors.

binations (*see* Fig. 6). Some of the amino acids (identified here by their three letter abbreviations) are specified by single codons (AUG, for Methionine), while others are specified by as many as six different codons (Leucine, for example).

The mRNA

Messenger RNAs may contain one open reading frame (monoscistronic), bounded by initiation and termination codons, or several (polycistronic). The mRNAs often have a nontranslated leader sequence at their 5' end preceding the initiation codon. Some mRNAs have short regions of four to six bases, which are complementary to sequences in the 3' end of 16S RNA exposed on the platform of the small ribosomal subunit. These sequences, called Shine-Dalgarno sequences, lie 10 or so bases upstream of the initiation codon, serving to bind and orient the mRNA, placing the initiation codon within the decoding center on the small subunit. The greater the region of complementarity, the stronger the binding and consequently the particular mRNA will be translated more frequently, relative to those with weaker Shine-Dalgarno interactions.

The Transfer RNAs (tRNAs)

All organisms use tRNAs as so called *adaptor* molecules to convert the sequence of nucleic acid codons into a sequence of amino acids. These are a family of small RNA molecules, 70–80 nucleotides long, which fold into a series of stem-loop structures, usually depicted as a characteristic cloverleaf structure (*see* Fig. 7). They typically contain unusual or modified bases, such as dihydrouracil (prominent in the D loop, so named because of the dihydrouracil content), pseudo uracil, and ribothymidine. The sequences of the various tRNAs are different, but they contain an invari-

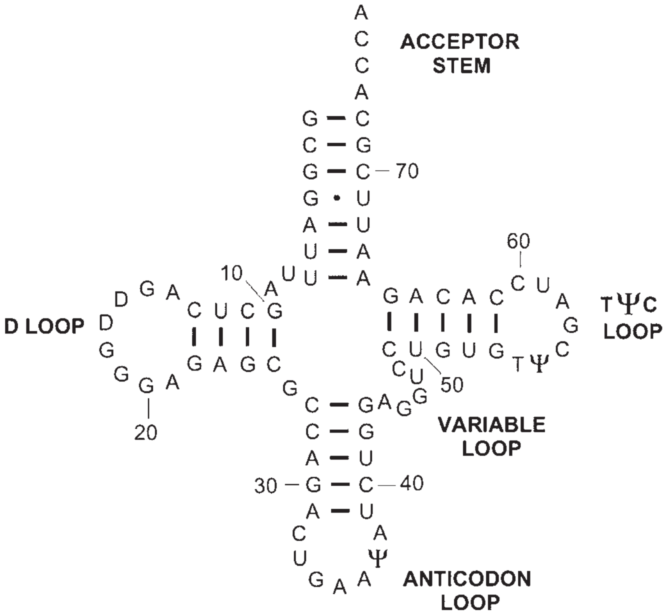


Fig. 7. Transfer RNA structure. Shown here is the secondary structure of phenylalanyl-tRNA (Phe-tRNA), in the familiar *cloverleaf* diagram characteristic of most tRNAs. The anticodon, 5'GAA, is complementary to one of the two codons for Phe (5'UUC) in Fig. 6. D, dihydrouridine; Ψ, pseudouridine; T, ribothymidine.

ant CCA, added postranscriptionaly at the 3' end of the amino acid acceptor stem, as well as a sequence, TΨCG, in the TΨC loop. As its name suggests, the variable loop is highly variable, both in sequence, as well as size. The anticodon and the amino acid that it specifies are separated by about 70 Angstroms, at opposite ends of the L-shaped molecule (*see* Fig. 8A) which forms when the D loop and TΨC loop fold further, stabilized by additional, tertiary base pair interactions.

The Aminoacyl-tRNA Synthetases

Amino acids are *activated* through coupling in ATP-dependent reactions to the appropriate tRNAs by amino acyl-tRNA synthetases specific for each amino acid and for the corresponding tRNAs. There are two general classes of synthetase enzymes, I and II, which tend to function as monomers or dimers, respectively. The enzymes recognize key identity elements at various locations in the cognate tRNAs.

The Ribosome

The small subunit (30S) of prokaryotic ribosomes contains one 16S RNA molecule and about 21 proteins. The RNA molecules are generally characterized by their sedimentation coefficient (S) as an indication of their size or mass. The larger and more compact the RNA molecule is, the higher the sedimentation coefficient. The 16S rRNA (ribosomal RNA) of *E. coli* is1542 nucleotides long (*see* Fig. 9), and the molecule is folded into a rather compact shape (*see* Fig. 10A). Adjacent complementary regions along the molecule result in the formation of a series of stem-loop structures. This has the effect of bringing together distant complementary regions to form additional stems,

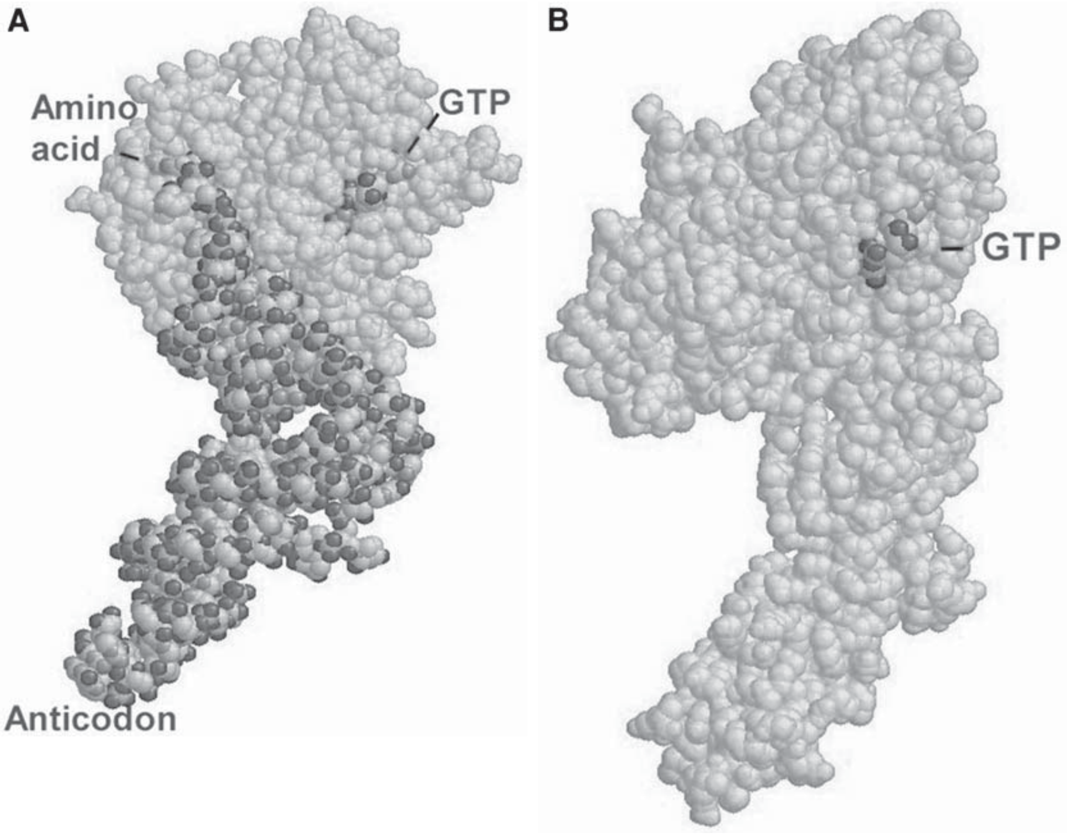


Fig. 8. Elongation factors EF-Tu and EF-G. **(A)** EF-Tu (lighter gray) is shown as a ternary complex with bound amino acyl-tRNA and GTP. **(B)** EF-G resembles the complex of the protein EF-Tu and amino acyl-tRNA. This is an example of macromolecular mimicry where similar binding sites accommodate the different species.

resulting in the formation of distinct domains through these long distance interactions (*see* Fig. 9). The 5' domain, extending from nucleotides 1 through 556 and the penultimate stem (positions 1400–1500) comprise the RNA structure of the *body* (*see* Fig. 10A). The 5' central domain (positions 567 to 883) forms the *platform*, and the *head* contains nucleotides 920–1372. The 21 proteins are distributed around the periphery, mainly on the back or solvent side. Only one protein, S12, is located on the interfacial side, near the decoding center.

The large subunit (50) contains two rRNA molecules, a 5S RNA, forming the central crown structure (*see* Fig. 10F), and a larger 23S RNA, forming the body of the subunit. The 33 proteins of the large subunit are located mainly on the periphery and the backside of the subunit. One of the ribosomal proteins, L12, is present in four copies. These proteins form two L12-dimer structures (*see* Fig. 10F). The L12 dimer stalk is the site where the elongation factors (EF-Tu and EF-G) first engage the ribosome. The 30S and 50S subunits associate during the initiation phase of protein synthesis to form a roughly spherical 70S ribosome.

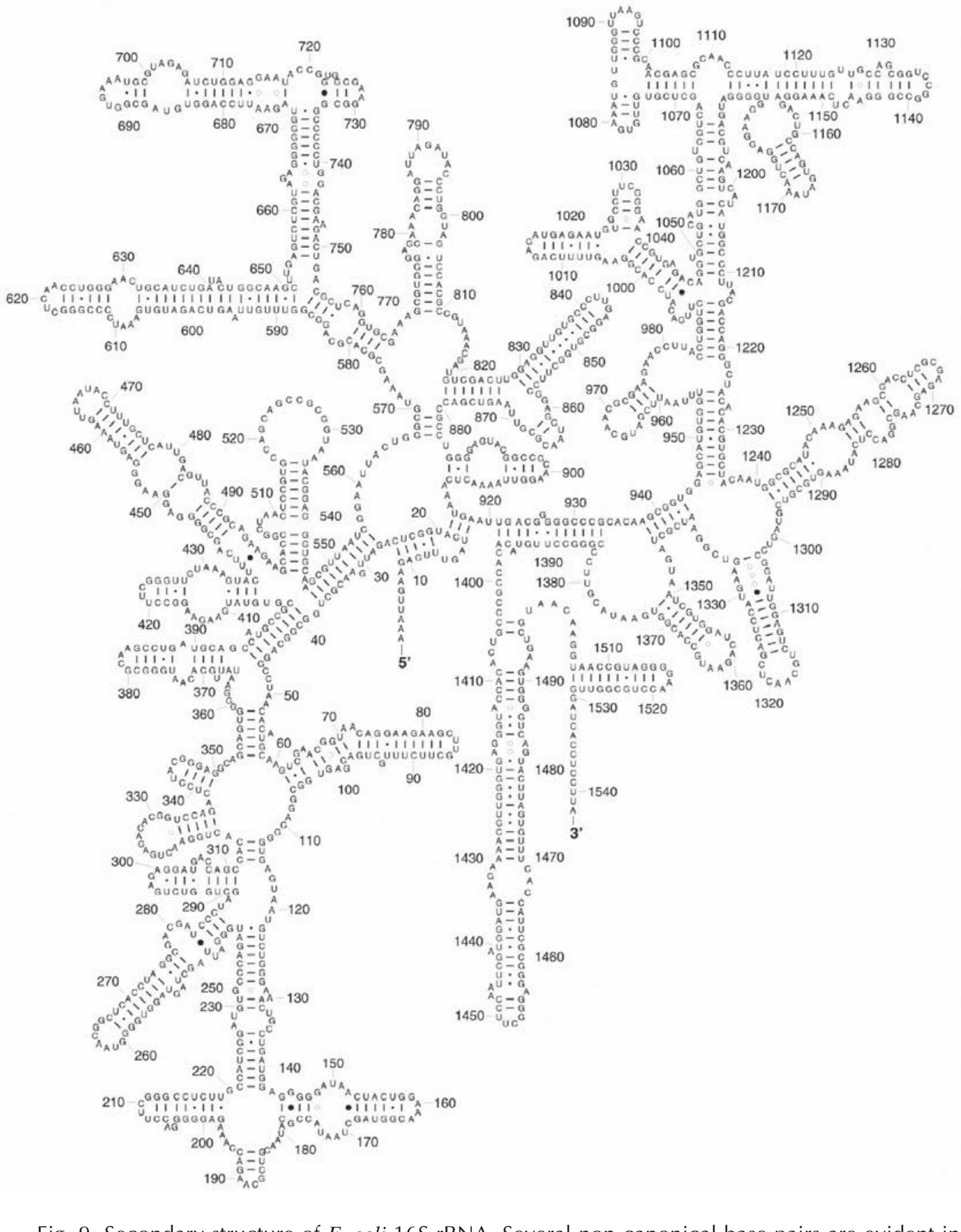


Fig. 9. Secondary structure of *E. coli* 16S rRNA. Several non-canonical base pairs are evident in the molecule, including A-G, G-G, U-G and U-U base pairs. The 16S RNA folds into a compact structure which defines the overall shape of the 30S subunit (*see* Fig. 10).

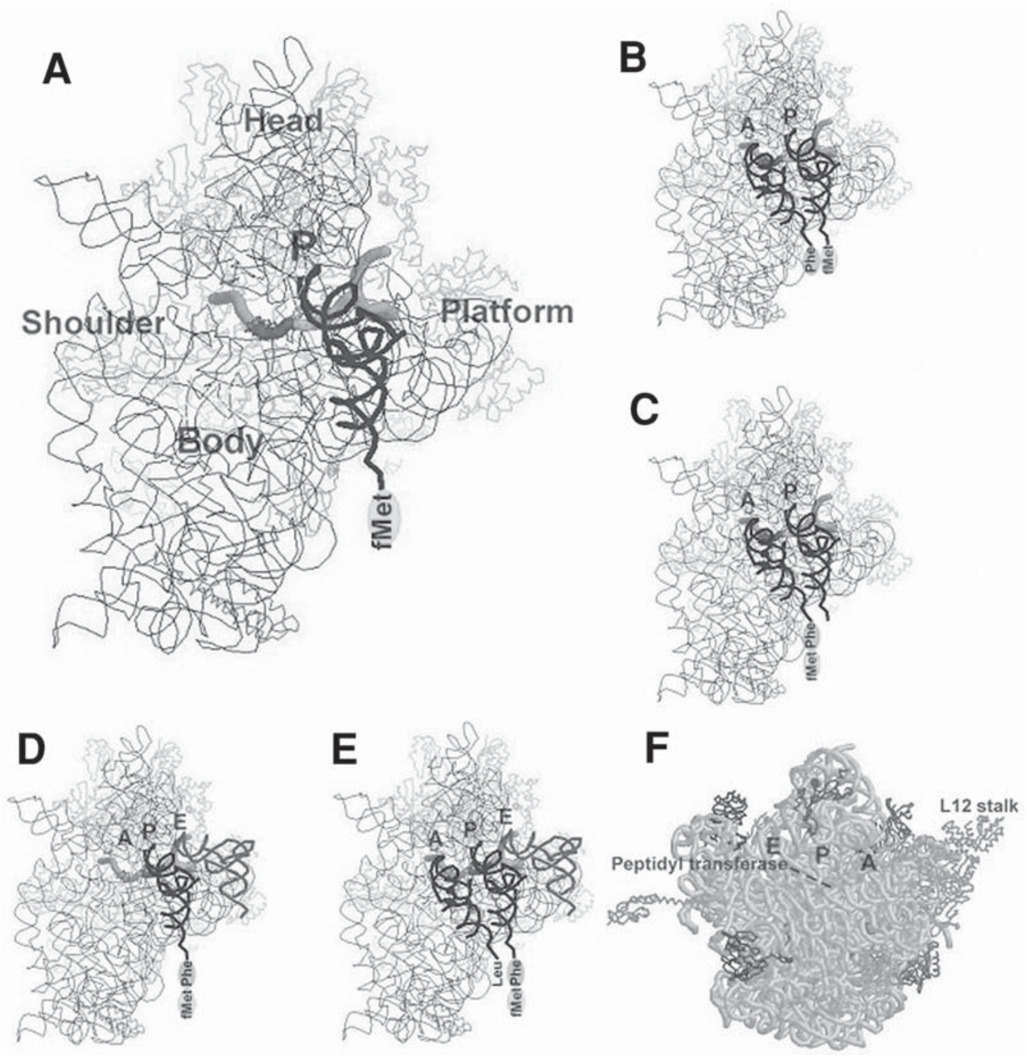


Fig. 10. Structure of the small and large subunits of the bacterial ribosome. **(A)** Interfacial aspect of the 30S subunit, showing the 16S RNA backbone (thin dark tracing) and associated proteins (alternatively shaded). A section of bound mRNA (thick bright backbone) includes the Shine-Dalgarno region (5' helical structure on the right), the initiation codon and an adjacent Phe codon. The anticodon of the initiator tRNA, fMet-tRNA (thick dark backbone) is base paired with the initiation codon in the P site. **(B)** Same as (A) with the addition of Phe-tRNA at the adjacent codon in the A site. **(C)** Peptidyl transferase catalyzed transfer of the fMet to the Phe-tRNA, resulting in a dipeptidyl-tRNA in the A site. **(D)** Post translocation state of the ribosomal small subunit. Through the action of EF-G with bound GTP, the uncharged initiator tRNA, fMet-Phe-tRNA and the mRNA are moved, in register, into the E and P sites. **(E)** Entry of the third amino acyl-tRNA into the A site, in advance of the ejection of the E site tRNA and peptide transfer (peptidyl transferase reaction). **(F)** Interfacial aspect of the 50S subunit, showing the 5S and 23S RNA (thick backbone) and associated proteins (variously shaded). The peptidyl transferase activity lies deep within a cleft in the 23S RNA, an area entirely devoid of protein. The coordinates for the 30S subunit, mRNA fragment, A, P, and E site tRNAs are available at the RCSB data base with PDB accession numbers 1GIX and 1JGO.

Initiation of Protein Synthesis

Initiation of protein synthesis takes place on the small subunit. The mRNA binds in the region between the head and the body, tracking from the platform through a downstream entrance channel formed between the head and the shoulder of the subunit (*see* Fig. 10A). If the mRNA has a Shine-Dalgarno sequence, it binds more efficiently, stabilized by base pairing with the complementary region of 16S rRNA exposed on the platform. This interaction places the initiation codon in the vicinity of the decoding center, where it can base pair with the anticodon of the initiator tRNA. The initiator tRNA in prokaryotic systems is fMet-tRNA, a special tRNA that carries a methionine modified by formylation on its amino group. This fMet-tRNA is escorted to the small subunit in a ternary complex with initation factor 2 (IF-2) and GTP. The fMet-tRNA is the only amino acyl-tRNA to bind directly and initially to the P site, where the initation codon will be able to form base pairs with the anticodon (*see* Fig. 10A). This interaction is facilitated by two other initiation factors, IF-1 and IF-3. IF-1 is a protein mimic of tRNA and it binds to the A site on the small subunit, forcing the IF-2 ternary complex to enter at the adjacent P site. IF-3 binds between the head and the platform, where it serves a dual role. It acts as an anti-association factor, ensuring a supply of free 30S subunits for initiation complex formation and it promotes the fidelity of initiation complex formation by favoring the dissociation of noncanonical complexes, such as base pair interactions between elongator tRNAs and the initiation codon. Upon proper recognition of fMet-tRNA by the initiation codon in the P site, the IF-2-bound GTP is hydrolyzed to GDP, causing a conformational change allowing IF-2-GDP to leave after depositing the initiator tRNA in the P site. The 50S subunit then joins the 30S initiation complex to form the 70S initiation complex, setting the stage for the elongation phase.

Elongation Phase of Protein Synthesis

The elongation factor EF-Tu (*see* Fig. 8A) escorts the various amino acyl-tRNAs to the ribosomal A site as ternary complexes with GTP. They engage the ribosome first at the L12 stalk and if the codon exposed in the A site (UUC, in Fig. 10B, for example) is complementary to their anticodon, the amino acyl-tRNA is deposited and the EF-Tu-bound GTP is hydrolyzed, favoring the exit of EF-Tu-GDP. Note that the mRNA is *kinked* between the adjacent codons in the decoding center. This kinking allows the A and P site tRNAs to read adjacent codons. In this manner the amino acyl ends of the A and P site tRNAs are placed within 5 Angstroms of each other (*see* Fig. 10B), in the peptidyl transferase center of the 50S subunit (*see* Fig. 10F) where they await peptide bond formation. Peptidyl transferase activity of the 50S RNA catalyzes the transfer of the fMet to the Phe-tRNA in the A site (*see* Fig. 10C). At this point in the elongation cycle the peptidyl-tRNA (fMet-Phe-tRNA, in the example of Fig. 10C) and the P site tRNA must be moved to open the A site for entry of the next amino acyl-tRNA able to base pair with the third codon brought into the A site. Elongation factor G, EF-G with bound GTP (*see* Fig. 8B) engages the L12 stalk on the large subunit and proceeds to carry out this translocation event, while maintaining the mRNA reading frame fixed by the A and P (E) site tRNAs. During translocation, the uncharged tRNA is moved into the E site while the A site peptidyl-tRNA moves into the P site (*see* Fig. 10D). EF-G is a striking example of tRNA mimicry by protein. Note the structural resemblance of EF-G (*see* Fig. 8B) and EF-Tu-amino acyl-tRNA (*see* Fig. 8A) which bind to similar sites on the ribosome. After hydrolysis of the bound GTP, EF-G leaves to make the A site available for entry of the next amino acyl-tRNA (*see* Fig. 10E). The growing peptide chain will emerge through a tunnel traversing the body of the 50S subunit. Upon entry of the next cognate amino acyl-tRNA the affinity of the E site for the uncharged tRNA is lowered, allowing for release of the E site tRNA. This cycle is repeated until a termination codon appears in the A site, setting the stage for the termination of protein synthesis.

Termination of Protein Synthesis

Termination codons exposed in the A site are recognized by protein release factors. RF1 recognizes UAA and UAG, and RF2 recognizes UAA and UGA. After binding of either release factor in the A site, peptidyl transferase activity transfers the C-terminal residue of the polypeptide to a water molecule, resulting in release of the protein. The departure of RF1 and RF2 is facilitated by a third release factor, RF3, which is another example of mimicry in translation systems. RF3 is a GTP binding protein that resembles EF-G.

Translation in Eukaryotes

The fundamental process of protein synthesis in eukaryotes resembles that of prokaryotes, with the introduction of additional complexity and regulatory features. The ribosomes are larger and the initiation factors are more numerous. Eukaryotes do not employ a Shine-Dalgarno-like mechanism to promote mRNA binding, but rather use a collection of initiation factors that recognize a uniquely eukaryotic decoration of the 5' end of mRNAs, the 5'-methyl-G cap structures. In addition, most of the mRNAs are further decorated by a poly(A) tail at their 3' end, which recruits poly(A)-binding proteins (PABPs). The 40S ribosome small subunit contains 18S RNA. The large subunit contains three molecules of RNA, a large, 28S RNA, 5S RNA and a species unique to eukaryotes, a 5.8S RNA. The 80S ribosome contains about 80 different proteins, some of which are homologues of bacterial ribosomal proteins, but many of them unique to eukaryotes.

The initiation phase of protein synthesis employs at least 12 initiation factors, containing as many as 28 protein subunits. These protein factors interact with the initiator Met-tRNAi, the mRNA, the 40S and 60S subunits to promote formation of the 80S initiation complex. The factor names are preceded by "e" to denote their eukaryotic nature. The factor eIF1A is homlogous to the bacterial IF1 and it aids in positioning the initiation codon in the decoding center. The homolog of IF2, eIF2 escorts Met-tRNAi to the P site of the ribosomal 40S subunit. The large factor eIF3, which contains 11 protein subunits, binds to the platform region of the 40S subunit. The eIF4F factor is a heterotrimeric factor; in addition to its central component, eIF4G, its eIF4A component is an ATP-dependent RNA helicase, and its eIF4E subunit binds to the 5'cap structure on the mRNA. The versatile factor eIF4G serves as a *bridging* factor that binds to eIF4E, as well as the PABPs (on the poly(A) tail), which has the effect of bringing the 5' and 3' ends of the message together. The message is then brought to the 40S subunit when the *loaded* eIF4G binds to eIF3 already on the subunit. The initiation AUG codon is as many as 100 or more nucleotides downstream of the 5'cap structure. This necessitates an ATP-dependent scanning search for the downstream initiation codon, recognizable by its ability to base pair with the waiting anticodon of Met-tRNAi. After selection of the initiation codon, eIF5 is recruited to aid in the dissociation of eIF2-GDP, eIF3, and eIF4 factors in preparation for the joining of the 60S subunit to form the 80S initiation complex. Some mRNAs, such as the polycistronic genomic RNA of the poliovirus, have no 5'-methyl-G cap structures, but contain internal ribosome entry sites (IRES) where initiation of protein synthesis can take place without the need for the initiation factors required for most cellular mRNAs.

Acknowledgments

We thank Christine Mione, Sara Rodriguez-Jato, Wesley Brooks, and Paul Wright for generating all of the figures on transcription and RNA processing for this chapter.

Glossary and Abbreviations

Transcription

Activator Proteins that augment and/or stabilize formation of the PIC, often by binding to activator DNA sequences.

Anti-sense RNA RNA transcript that is complementary to a known coding RNA; transcribed from the non-template strand of DNA.

BRE TFIIB recognition element.

Closed binary complex The DNA-protein complex consisting of RNA polymerase (or the PIC in eukaryotes) and double-stranded DNA of the promoter where the double helix of DNA is not melted.

Coactivator A protein or protein complex that act as an intermediate or bridge for interactions between activators and the PIC.

Core enzyme Form of prokaryotic RNA polymerase that functions during transcription elongation; consists of two α subunits, one β subunit, and one β' subunit.

Core promoter *(also referred to as the basal promoter)* In eukaryotes, the minimal promoter region that is required to form the PIC and initiate transcription.

CTD C-terminal domain of the largest subunit of RNA polymerase II.

DNA template strand Strand of the DNA double helix that encodes an RNA.

DPE Downstream promoter element.

GTFs General transcription factors.

Hairpin A region of an RNA or single-stranded DNA, that folds back upon itself and forms intramolecular Watson-Crick base pairing.

Holoenzyme In prokaryotes, the core RNA polymerase enzyme plus sigma factor; in eukaryotes, a complex of RNA polymerase II, several general transcription factors (usually not including TBP), and the mediator co-activator complex.

Inr Initiator element; the DNA sequence spanning the transcription initiation site that is bound by initiator proteins that mediate formation of the PIC in TATA-containing or TATA-less promoters.

Mediator complex Co-activator complex that is bound to the CTD of RNA polymerase II.

Nucleosomal array A stretch of DNA organized into a series of nucleosomes forming a beads-on-a-string structure.

Nucleosome Basic unit of DNA compaction in eukaryotic nuclei; consists of ~200 bp of DNA wrapped around an octamer core of histone proteins.

Open binary complex The DNA protein complex of double-stranded DNA and RNA polymerase (or the PIC in eukaryotes) where the DNA duplex bound by the polymerase is melted into single-stranded regions.

PIC Pre-initiation complex.

Pol I, pol II, pol III RNA polymerase I, II, and III, respectively.

Promoter clearance *(also referred to as promoter release)* Release of the PIC from the promoter and conversion of the PIC to the transcription elongation complex.

Rho protein Protein that mediates rho-dependent transcription termination in prokaryotes.

RNA:DNA duplex A region in which one strand of DNA is base paired with its complementary sequence of RNA.

rNTP Ribonucleotide triphosphate.

rRNAs Ribosomal RNAs.

TAFs TBP-associated factors; subunits of TFIID.

TATA box A short region ~30 bp upstream of the transcription initiation site that contains the DNA sequence TATA and is bound by TBP.

TBP TATA-binding protein; the subunit of transcription factor TFIID that binds to the TATA box.

TFIIA, B, D, E, F, H General transcription factors that are components of the pre-initiation complex of genes transcribed by RNA polymerase II.

Transcription elongation complex RNA polymerase and its associated components and factors that forms after promoter clearance and elongates a nascent RNA.

Watson-Crick base pairing Pairing of complementary bases; A:T, A:U, G:C.

RNA Processing

Capping Addition of a 7-methyl G residue to the 5'end of mRNAs via a 5'→5' linkage.

Deamination Removal of an amine group from a base; deamination of cytosine yields uracil.

Editosome Macromolecular complex that carries out RNA editing.

Exon Portion of a gene (and RNA) that is included in the final fully-processed mature mRNA molecule.

Exon definition The process in RNA splicing that allows the spliceosome to accurately recognize and splice exons within an unspliced primary RNA transcript.

Intron DNA (and RNA) sequences located between exons that interrupt the mRNA-coding portion of a gene (and RNA transcript).

Polyadenylation Addition of a string of A residues to the 3' end of mRNAs in eukaryotes.

Polyadenylation signal sequence The AAUAA sequence that dictates downstream cleavage of the primary RNA transcript, thereby forming the 3' end of the mRNA and the site for polyadenylation.

Post-transcriptional Molecular processes (such as RNA processing) that occur after an RNA has been transcribed.

RNA editing The non-templated addition, deletion, or modification of nucleotides within a mature mRNA that occurs after transcription.

RNA splicing Removal of introns from the primary RNA transcript to form the mature mRNA.

Spliceosome Large RNA-protein complex that carries out RNA splicing reactions.

SR proteins Family of serine- and arginine-rich proteins that play central roles in RNA splicing.

Transesterification reaction Chemical reaction occurring during splicing that involves breakage or exchange of one ester bond for the formation of a new ester bond.

Translation

A site Ribosomal site where aminoacyl-tRNAs enter and bind during the elongation phase of protein synthesis.

Anticodon A series of three nucleotides in the base loop of tRNA that are complementary to a codon specifying the amino acid attached to a particular amino acyl-tRNA.

Codon A series of three bases (A, G, C or T) in the mRNA open reading frame that specify a particular amino acid.

E site Exit site, where uncharged tRNAs are bound after translocation, from which they exit the ribosome.

EF-G Prokaryotic Elongation factor G, the translocation factor for prokaryotic ribosomes.

EF-Tu Prokaryotic Elongation Factor T, forms ternary (three component) complex with amino acyl-tRNA and GTP; carries aminoacyl-tRNA to ribosomal A site.

IF2 Prokaryotic Initiation factor 2, the factor that binds, along with GTP, to initiator tRNA, and carries the initiator tRNA to the ribosomal P site.

IRES Internal Ribosome Entry Site; eukaryotic ribosomes can initiate protein synthesis at these sites in mRNAs that lack the 5' cap decoration.

P site Ribosomal site where initiator tRNA enters and binds.

Peptidyl-tRNA tRNA with the growing peptide attached at its 3' end.

Shine-Dalgarno sequence A short four-to-seven base sequence in the 3' end of bacterial small ribosomal subunit RNA (16S RNA) that is complementary to a short sequence of bases upstream of the initiation codon in some mRNAs.

Translocation EF-G catalyzed movement of tRNAs and mRNA in the ribosome following peptidyl transfer to the aminoacyl-tRNA in the A site. During translocation the uncharged tRNA in the P site and the peptidyl-tRNA in the A site, are moved, along with the mRNA, into the E and P sites, respectively.

Suggested Reading

Transcription and RNA Processing

Cramer, P., Bushnell, D. A, Fu, J., Gnatt, A. L., et al. (2000) Architecture of RNA polymerase II and implications for the transcription mechanism, Science 288, 640–649.

Cramer, P., Bushnell, D. A., and Kornberg, R. D. (2001) Structural basis of transcription: RNA polymerase II at 2.8 angstrom resolution, Science 292, 1863–1876.

deHaseth, P. L., Zupancic, M. L., Record, M. T. (1998) RNA polymerase-promoter interactions: The comings and goings of RNA polymerase, J. Bacteriol. 180, 3019–3025.

Hirose, H. and Manley, J. L. (2000) RNA polymerase II and the integration of nuclear events, Genes Dev. 14, 1415–1429.

Lee, T. I. and Young, R. A. (2000) Transcription of eukaryotic protein-coding genes, Annu. Rev. Genet. 34, 77–137.

Mooney, R. A., Artsimovitch, I., and Landick, R. (1998) Information processing by RNA polymerase: Recognition of regulatory signals during RNA chain elongation, J. Bacteriol. 180, 3265–3275.

Translation

Garret, R. A., Douthwaite,S. R., Liljas, A., Matheson, A. T., Moore, P. B., and Noller, H. F. (2000) The Ribosome: structure, function, antibiotics and cellular interactions, ASM Press, Washington, DC.

Gutell R. R. (1996) Comparative Sequence Analysis and the Structure of 16S and 23S rRNA, in: Ribosomal RNA. Structure, Evolution, Processing, and Function in Protein Biosynthesis, (Dahlberg, A. and Zimmerman, B. eds.) CRC Press, Boca Raton, FL, pp. 111–128.

Yusupova, G. Z., Yusupov, M. M., Cate, J. H., and Noller, H. F. (2001) The path of messenger RNA through the ribosome, Cell 106, 233–241.

Yusupov, M. M., Yusupova, G. Z., Baucom, A., Lieberman, K., Earnest, T. N., Cate, J. H., and Noller, H. F. (2001) Crystal structure of the ribosome at 5.5 A resolution, Science 292, 883–896.

Part II Molecular Genetics

A. Genomics

6 Epigenetic Mechanisms Regulating Gene Expression

John R. McCarrey

Epigenetics:

> " The study of mitotically and/or meiotically heritable changes in gene function that cannot be explained by changes in primary DNA sequence."
>
> A. D. Riggs, 1996

Introduction

Epigenetic mechanisms regulate gene function in a heritable manner, but do so without modulating the DNA sequence of the affected gene. Many different genetic functions are influenced by epigenetic mechanisms in various species. These include regulation of gene expression, DNA modification and restriction, genomic imprinting, X-chromosome inactivation, paramutation, position effect variegation, mating type, cell determination, transposable elements, and mutator and suppressor genes. This chapter will focus on epigenetic mechanisms that regulate gene expression and the manner in which they accomplish this in mammalian species.

Nuclear DNA acts as the repository of genetic information in eukaryotic cells. In mammals, and many other animal species, a complete representation of the genome is maintained in essentially every nucleated cell. However, only a subset of this collection of genes is expressed in any particular cell type. Thus, it is not the presence of specific genes, but rather the expression of specific genes, that leads to the unique identity and function of any particular cell.

For over 25 years, the mechanisms that regulate gene expression have been the focus of extensive investigation. For protein-encoding genes, two primary steps are involved in gene expression: transcription of DNA into RNA, and translation of that RNA into a polypeptide. This affords two levels of regulation of gene expression: transcriptional regulation and translational (or post-transcriptional) regulation. For tissue-specific genes (those expressed in only a subset of tissues or in a single tissue or cell type), regulation is primarily manifest at the transcriptional level. Extensive studies of this process have revealed a consensus mechanism whereby the promoter region, typically located at the 5'-end of the gene, acts to bind specific proteins called transcription factors which, in turn, attract (or prevent) binding of the RNA polymerase that is required to initiate transcription.

Binding of transcription factors to specific gene promoters and to specific sites within those promoters is regulated by the ability of a DNA binding domain within each protein factor to recognize a unique three-dimensional structure of double-stranded DNA. This unique structure is imparted by a specific nucleotide sequence, typically 5–15 base pairs (bp) in length. Thus, this mechanism does rely on the DNA sequence and is therefore not a truly epigenetic mechanism. However, because these transcription factors can be either ubiquitous or tissue-specific, and can either promote or inhibit transcription, this mechanism can modulate tissue-, cell-type, or developmental-stage specificity of transcription, as well as controlling the relative level (or frequency) of transcription. Nevertheless, protein-DNA interactions between transcription factors and promoter sequences, respectively, are not the only mechanism by which gene expression is regulated in eukaryotic cells.

In mammals there are several examples in which genes are regulated by mechanisms other than transcription factors. For example, in female somatic cells, genes on the active X chromosome are transcribed, whereas homologous genes on the inactive X chromosome remain transcriptionally silenced. This is despite the fact that both the active and inactive copies of these genes share identical nucleotide sequences and reside within the same nucleus. Thus the presence of identical promoter sequences and cognate transcription factors alone does not ensure identical regulation of genes. Similarly, in mammals, the phenomenon of genomic imprinting results in the expression of only one of the two copies of a particular gene within a single diploid cell. In this case the choice of which allele is expressed is dictated by the parental origin of that allele. However, the mechanism that regulates such monoallelic expression cannot be based solely on transcription factors and promoter sequences, because the former are present throughout the nucleus in which both alleles reside, and the latter are often identical on both alleles.

The unavoidable conclusion from these observations is that there must be additional mechanisms by which gene expression is regulated in eukaryotic cells, and these mechanisms must function in a manner that does not depend on differences in nucleotide sequence or the cell-type specific presence or absence of transcription factors. Yet, as exemplified by the examples noted earlier for X-chromosome inactivation and genomic imprinting, these mechanisms must function in a heritable manner, such that the same alleles remain expressed or silenced, even after replication of the DNA and division of one cell to produce two daughter cells.

We now know that there are multiple mechanisms that meet the criteria of epigenetic mechanisms in that they regulate gene expression in a heritable manner that does not rely on differences in DNA sequence. Examples of mechanisms that either have been shown to operate in this manner or have the potential to operate in this manner include: DNA methylation, chromatin structure and/or composition, DNA loop domains and association with the nuclear matrix, and DNA replication timing.

DNA Methylation

In mammals, methylation of DNA is found only on cytosines present in a 5'-CpG-3' dinucleotide sequence. Because cytosine and guanine are complementary bases, wherever there is a CpG dinucleotide in one DNA strand, there will be a complementary CpG on the opposite strand. This double-stranded structure can exist in three different states with respect to methylation (*see* Fig. 1). It can be *fully methylated*,

See companion CD for color Fig. 1

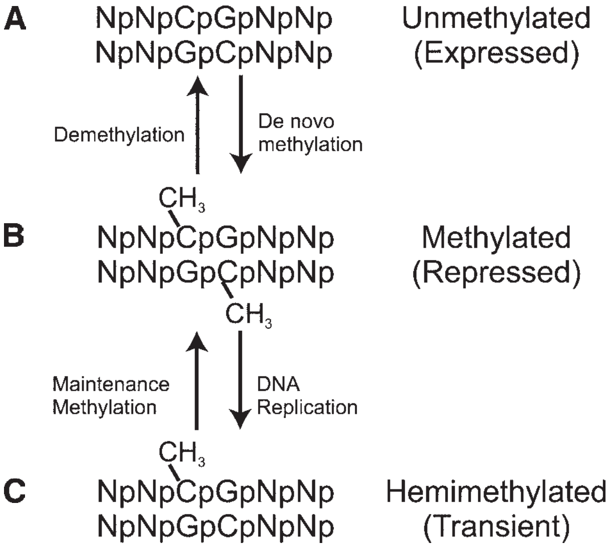


Fig. 1. Alternate states of DNA methylation in mammalian DNA. Methylation occurs only on cytosines present in CpG dinucleotides in mammalian DNA. A CpG dinucleotide sequence in one DNA strand mandates the presence of a complementary CpG dinucleotide in the other strand of double-stranded DNA. **(A)** If both cytosines in such a site are unmethylated, the site is said to be completely unmethylated. This structure is often found in actively expressed or potentiated genes, especially in the 5' regulatory region. **(B)** An unmethylated site can undergo *de novo* methylation to form a fully methylated site in which both cytosines are methylated. This structure is often found associated with repressed genes. Conversely a demethylase activity can convert a fully methylated site to a fully unmethylated site in the absence of DNA replication. **(C)** Upon semi-conservative replication of a fully methylated site, a hemimethylated site is formed. This structure is typically transient as a maintenance DNA methyl transferase rapidly recognizes a hemimethylated site and returns it to a fully methylated state. The function of the maintenance methylase provides a mechanism to heritably maintain DNA methylation patterns. C, cytosine; G, guanine; N, any base; p, phosphate bond; CH_3, methyl group.

meaning that both cytosines are methylated (*see* Fig. 1B), or it can be completely unmethylated if neither cytosine is methylated (*see* Fig. 1A). When a fully methylated site is replicated by semi-conservative replication, the resulting structure is hemimethylated (*see* Fig. 1C). This is typically a transient state because it forms a template for a ubiquitously functioning DNA maintenance methyl transferase that recognizes the hemimethylated structure and returns it to a fully methylated state. Thus, fully methylated and unmethylated sites are maintained (and/or re-established) throughout replication of DNA and cellular division. In this way methylated and unmethylated states of DNA are heritable.

It is possible for an unmethylated site to be directly converted to a fully methylated site, and vice versa. Methylation of an unmethylated site is achieved by a *de novo* methylase, whereas a direct transition from a fully methylated to an unmethylated structure in the absence of DNA replication is accomplished by a demethylase activity. The function of these enzymatic activities and the manner in which they are regulated are not as well-characterized as that of the maintenance methylase activity.

However, there is ample evidence that such activities do indeed exist. Shortly after fertilization in the mouse, nearly all of the methylation that is brought into the zygote by the gametic genomes is lost, such that the blastocyst genome is nearly devoid of DNA methylation except for that at a few imprinted sites. This may occur either by dilution of methylated strands as replication proceeds in the absence of maintenance methylase activity or by direct demethylation, or by some combination of these two mechanisms. Subsequently, at about the time of gastrulation, there is a *de novo* methylation event at numerous different sites throughout the genome. This must be accomplished by a *de novo* methylase because completely unmethylated sites become fully methylated. Following gastrulation, many different cell lineages become allocated and begin to develop and differentiate. Coincident with this, selective demethylation of many tissue-specific genes is often observed within the cell lineage in which these genes will ultimately be expressed. In most cases this appears to occur via a demethylase activity, because in at least some cases demethylation occurs in the complete absence of DNA replication or cellular division.

In addition to tissue-specific genes that are expressed in a limited tissue-, cell-type, or developmental-stage specific pattern, another set of *housekeeping* genes is widely expressed in a ubiquitous and constitutive manner. These genes, which do not require as complicated transcriptional regulation as that needed for tissue-specific genes, often bear a *CpG island*, most commonly in the 5'-portion of the gene. A CpG island has been defined as a region in the mammalian genome of >100 bp with a GC content of >50% that lacks the typical underrepresentation of CpG dinucleotides seen in other regions of the genome. Generally, CpG islands remain constitutively unmethylated throughout development and differentiation of cells. Exceptions to this rule include CpG islands associated with genes on the inactive X chromosome or with nonexpressed, inactive alleles of imprinted genes, as well as those associated with certain genes in cancerous tumors (e.g., tumor-suppressor genes). In these cases, the island associated with the nonexpressed allele or gene is typically methylated.

For both individual CpG dinucleotides located in non-CpG island regions, and CpG dinucleotides within CpG islands, a general correlation has been observed between the presence of DNA methylation and inhibition of expression, and between the absence of DNA methylation and active transcription. This is especially true for sites in the 5'-flanking region or in the 5'-half of transcribed portions of genes. At least two types of mechanisms have been proposed by which DNA methylation, or the lack thereof, might contribute to regulation of transcription. In one case, the presence or absence of methylation on key cytosines within a particular transcription factor binding site may modulate the ability of the factor to bind to that site. In a second scenario, the presence or absence of methylation at sites either within factor binding sites or in regions adjacent to factor binding sites may inhibit factor binding indirectly by affecting chromatin structure.

A direct mechanistic connection has now been established between DNA methylation and chromatin structure (*see* "Interactions Among Epigenetic Mechanisms"). In this case, it is suggested that the presence of methylation stabilizes a condensed (*closed*) chromatin structure that is, in turn, refractory to binding by transcription factors and/or RNA polymerase. Conversely, an absence of methylation leads to a less condensed (*open*) chromatin structure that is accessible to transcription factors and RNA polymerase. Effects of DNA methylation on chromatin structure appear to be mediated by methylated DNA-binding proteins that bind to methylated DNA on the basis of the presence or absence of methylation, rather than on the basis of a particular binding sequence as is the case for transcription factors.

Two methylated-DNA binding proteins were originally identified, MeCP1 and MeCP2. MeCP1 is a large protein complex that binds best to regions of DNA containing >10 methyl-CpGs, and has been shown to be involved in repression of transcription from densely methylated promoters. It also binds to, and represses transcription from, more sparsely methylated promoters, although this is a much weaker interaction. MeCP2 is a single polypeptide that can bind to as few as a single fully methylated CpG site. In vivo it appears to bind predominantly to highly methylated satellite DNAs adjacent to centromeres in the mouse genome, but shows a more dispersed binding pattern in the genomes of humans and rats, which do not contain highly methylated satellite DNA.

Recently, screens for cDNAs encoding methyl-CpG binding domains (MBDs) have revealed at least four such genes, *MBD1-4*. The MBD1 protein is a component of the MeCP1 protein complex. MBD2-*4* encode methylated DNA binding proteins that are distinct from those associated with either MeCP1 or MeCP2, but bear a striking similarity to the MBD of MeCP2. The products of *MBD2* and *MBD4* bind to methylated CpGs both *in vitro* and *in vivo*, and are thus considered to be additional candidates for mediators of mechanisms associated with methylated DNA.

The manner in which tissue-, cell-type and/or gene-specific patterns of DNA methylation are established or modulated remains to be fully elucidated. However there appears to be a combination of general and specific mechanisms that contribute to this process. The general mechanisms include those that result in genome-wide loss or gain of methylation, especially during early embryogenesis, along with the maintenance methylase activity that re-establishes full methylation at hemimethylated sites following replication of DNA. Cell-type, developmental-stage, and gene-specific demethylation have been shown to be regulated by signal sequences within the promoter region of at least one tissue-specific gene. Regulation of CpG island methylation has also been shown to be dependent on signal sequences within certain imprinted genes. However, it appears that different methylases and demethylases may be responsible for *de novo* methylation/demethylation of CpG dinucleotides within or exclusive of CpG islands, respectively.

Chromatin Structure and Composition

See companion CD for color Fig. 2

Although DNA exists in a double helix structure within eukaryotic cells, this structure alone, otherwise known as *naked DNA*, is rarely found in cells in vivo. Rather the nuclear DNA is typically complexed with proteins to form chromatin. At a primary level, chromatin structure commonly involves double-stranded DNA wrapped periodically around protein structures called nucleosomes (*see* Fig. 2A). Approximately 150 bp of double-stranded DNA are wrapped in two superhelical turns around each nucleosome, and nucleosomes are typically separated by approx 10 bp of double-stranded DNA. This forms what has been termed a *beads on a string* structure that is approx 10 nm in diameter. This is also known as an *open* or *potentiated* chromatin structure, and is the structure most commonly found in genes that are undergoing active transcription. In this structure the nucleosomes are typically evenly spaced at regular intervals of approx 10 bp, however, as necessary, nucleosomes can be displaced, eliminated, or newly formed to facilitate initiation of transcription from a

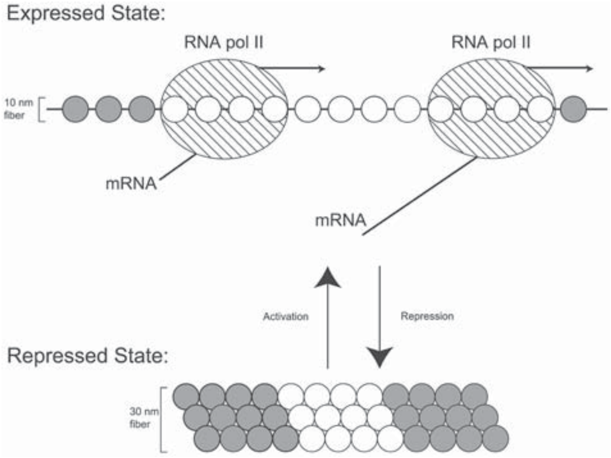


Fig. 2. Alternate states of chromatin structure. **(A)** In eukaryotic cells double-stranded DNA is typically complexed with clusters of histones called nucleosomes to form a *beads on a string* structure. This structure, which forms a fiber of approx 10 nm in diameter, is typically found in genes that are undergoing active transcription as shown in the Expressed State in this figure. The RNA polymerase II complex is able to traverse and transcribe portion of this structure to produce mRNA transcripts. **(B)** Long-term repression of gene transcription is accomplished by condensation of chromatin to form a 30 nm fiber as shown in the Repressed State in this figure. This condensed structure is refractory to binding by transcription factors and/or RNA polymerase II inhibiting initiation of transcription. Open circles, nucleosomes within the transcribed portion of the gene; filled circles, nucleosomes between genes; hatched oval, the RNA II polymerase complex.

promoter region. Such modifications may be required to permit or even promote access to specific factor binding sites in the double-stranded DNA.

Changes in chromatin structure at the level of nucleosome positioning can be modulated by ATP-dependent chromatin remodeling complexes. The SWI2/SNF2 complex originally discovered in *Drosophila*, is the best characterized of these. It is also conserved in mammals where it functions in a similar manner. The primary function of chromatin remodeling machinery is to remodel nucleosomal arrays to enhance accessibility to transcription factors. Once the necessary transcription factors and then the RNA polymerase complex have become bound to the gene promoter, transcription can proceed through the entire gene as it exists in the 10 nm diameter chromatin structure.

An alternate chromatin structure is achieved when the 10 nm chromatin fiber becomes supercoiled to form a 30 nm structure sometimes referred to as a *solenoid* (*see* Fig. 2B). This structure is typically found in transcriptionally repressed genes. This highly condensed structure is refractory to nucleosome displacement and/or factor binding. Thus, if transcription factors cannot gain direct access to their cognate binding sites in the double-stranded DNA, they will not bind and hence will not promote initiation of transcription. The option to exist in this repressive chromatin structure reconciles how a gene that contains all necessary factor-binding sites can reside in a nucleus in which all the necessary transcription factors are present, and still not be transcribed.

The composition of chromatin has been shown to vary in a manner that correlates with the different structures described earlier. A primary component of nucleosomes are histones. Each nucleosome consists of an octamer of two molecules each of histones H2A, H2B, H3 and H4, plus one linker histone H1 or H5. When chromatin is present in the 10-nm open or potentiated configuration most often associated with active transcription, the histone H4 in that region is often hyperacetylated. Conversely, when chromatin is present in the 30 nm repressed structure, histone H4 is typically hypoacetylated. Acetylation of histone H4 is believed to inhibit condensation of the 10 nm chromatin fiber and thus contribute to the maintenance of an open or potentiated chromatin structure, whereas the absence of acetylation on histone H4 contributes to chromatin condensation to form the repressive 30 nm structure. The acetylation of histones is regulated by a dynamic balance of histone acetyl transferases (HATs) and deacetylases (HDACs). The recruitment of these enzymes to specific loci can be regulated by transcription factors located by specific protein-DNA interactions, or by the methylated DNA-binding proteins. The latter are particularly associated with histone deacetylases. This provides a mechanistic basis for the frequently observed correlation between hypermethylation of DNA and a condensed, repressed chromatin structure (*see* "Interactions Among Epigenetic Mechanisms").

DNA Loop Domains and Association with the Nuclear Matrix

Regulation of chromatin structure and transcriptional activity can occur over relatively large distances within the mammalian genome. While this affords significant advantages for coordinate regulation of gene expression, it also poses potential disadvantages in that it raises the possibility that controlling influences targeted to one locus could inadvertently affect other loci, resulting in ectopic and/or inappropriate gene expression or suppression. This potential problem is mitigated in mammals by an additional level of regulation that results in segregation of chromatin into essentially independent loop domains (*see* Fig. 3). Delineation of these domains is believed to be achieved by the presence of *boundary elements* and/or by anchorage of the boundaries of each loop to a three-dimensional proteinaceous structure in the nucleus called the *nuclear matrix*. This arrangement affords multiple additional opportunities for control of gene expression. First, each loop is essentially insulated from adjacent loops and can thus be regulated independently. Second, it has been proposed that many factors and enzymes that regulate DNA replication, transcription, and post-transcriptional processes may be embedded in the nuclear matrix, so that proximity of individual genes to the matrix may influence the rate at which these processes occur in a gene-specific manner.

See companion CD for color Fig. 3

Specific sequences in chromosomal DNA have been shown to have nuclear matrix-binding capacity. These regions are known as *matrix attachment regions* or *MARs* (the nuclear matrix is also known as the *nuclear scaffold* and so these attachment regions are also known as *scaffold attachment regions* or *SARs*). MARs/SARs can be A/T-rich sequences that are often found at the boundaries of transcription units or in the vicinity of transcriptional enhancers. There appear to be at least two classes of MARs, constitutive and facultative. Constitutive MARs are believed to remain bound to the matrix in all cell types at all developmental stages. Thus, constitutive MARs would define a primary loop domain structure. Facultative MARs are believed to be differentially associated with, or disassociated with the matrix in a gene-, cell-type, and developmental-stage specific manner. Thus, facultative MARs

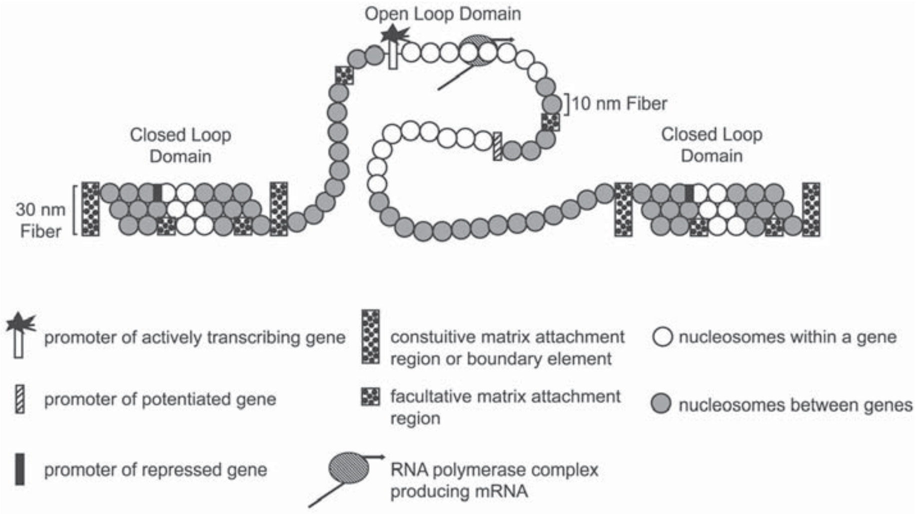

Open Loop Domain
10 nm Fiber
Closed Loop Domain
Closed Loop Domain
30 nm Fiber
promoter of actively transcribing gene
constuitive matrix attachment region or boundary element
nucleosomes within a gene
promoter of potentiated gene
facultative matrix attachment region
nucleosomes between genes
promoter of repressed gene
RNA polymerase complex producing mRNA

Fig. 3. Organization of DNA loop domains. Adjacent chromatin loop domains can exist in alternate states of condensed (closed) or decondensed (open) structure. Genes within closed domains are typically repressed. Genes within open domains are potentiated for transcriptional activation. Activation of transcription requires transcription factor binding and initiation of RNA synthesis by RNA polymerase II. (Modified with permission from Krawetz, et al. [1999]).

would have the potential to form a secondary loop structure, and/or to regulate tissue- or stage-specific associations between individual genes and the nuclear matrix. The extent to which boundary elements and MARs represent identical or distinct structures remains to be clearly elucidated.

Chromosome loop domains show both developmental-stage and tissue-specificity. These loops are generally small in nuclei of sperm and early embryonic cells, but become larger in differentiating somatic cells. Although the germ cells also undergo significant differentiation during gametogenesis, the loop sizes appear to remain generally smaller than in somatic cells. This difference in loop size could be produced by differential activity of facultative MARs, such that more of these are functional in germ cells and early embryonic cells leading to an increased incidence of DNA-matrix interactions in these cells and hence, delineated loops of smaller average size. It is tempting to suggest that small loop size is characteristic of a state of genetic/developmental pluripotency. This is consistent with the observation that genes that are actively transcribed, or in a state of readiness or *potentiated* for transcriptional activation and are typically found more closely associated with the nuclear matrix than are transcriptionally repressed genes. Thus, in the gametes and early embryonic cells, maximum use of facultative MARs could maintain a maximum number of genes in a potentially expressible state. However, as specific cell lineages become committed to a particular differentiated phenotype, many unneeded genes could become terminally repressed by selective disassociation of facultative MARs from the matrix to sequester the repressed genes away from transcription factors and RNA polymerase complexes associated with the matrix.

Replication Timing

The entire genome is duplicated during S phase of the cell cycle. However, the order in which gene loci are replicated varies among cell types and at different developmental stages. Typically, genes that are actively expressed in a particular cell type are replicated relatively early during S phase, whereas repressed genes are replicated later. For example, the same tissue-specific gene that is replicated early in the expressing cell type may be replicated later in nonexpressing cell types.

For most genes, both alleles are replicated simultaneously during S phase. However, in cases of monoallelic expression as seen for imprinted genes or genes on the active and inactive X chromosomes, the two alleles replicate asynchronously. In the case of X-linked genes, expressed alleles on the active X chromosome tend to be replicated earlier than their nonexpressed homologues on the inactive X. Interestingly, while imprinted genes also show asynchronous replication, the paternal allele is always replicated earlier than the maternal allele, regardless of which allele is expressed and which is repressed.

Differential replication timing has been demonstrated by direct fluorescence *in situ* hybridization (FISH) and by DNA hybridization using cells separated according to DNA content by fluorescence-activated cell sorting (FACS). It is not clear whether this is a cause or effect of related epigenetic mechanisms. It has been suggested that key chromatin remodeling and/or transcription factors may be limiting within individual nuclei and that following disassociation of these factors from the DNA during replication, early replicating genes or alleles might be afforded a preferential opportunity to reassociate with these factors immediately following replication (*see* Fig. 4). If so, this could then contribute to subsequent differential transcriptional activity of these genes or alleles. However, this theory is not supported by the observation that for certain imprinted genes, the maternal allele is preferentially expressed even though the paternal allele is replicated early. An alternative hypothesis is that early replication might simply be a reflection of the presence of DNA replication enzymes associated with, and/or embedded in the matrix, since actively expressed or potentiated genes tend to be more closely associated with the nuclear matrix. In either case, replication timing is indicative of epigenetic differences among gene loci and/or between alleles of the same gene.

See companion CD for color Fig. 4

Interactions Among Epigenetic Mechanisms

The epigenetic mechanisms discussed earlier represent additional levels of gene regulation that are available to mammalian cells beyond that afforded by direct protein-DNA interactions between transcription factors, the RNA polymerase complex, and gene promoters. An important question is how these multiple levels of gene regulation are orchestrated by the cell to achieve proper patterns of expression of batteries of housekeeping and tissue-specific genes. Clearly a variety of strategies are employed depending on the particular regulated gene and/or the particular cell type in which the regulation takes place. An example cascade of epigenetic and genetic mechanisms is presented below to demonstrate how these different mechanisms could interact to facilitate transcriptional activation of a tissue-specific gene.

See companion CD for color Fig. 5

Activation of transcription of a tissue-specific gene requires a derepression process that takes the gene from a transcriptionally repressed state first to a potentiated state, and subsequently to a transcriptionally active state (*see* Fig. 5). The transcriptionally

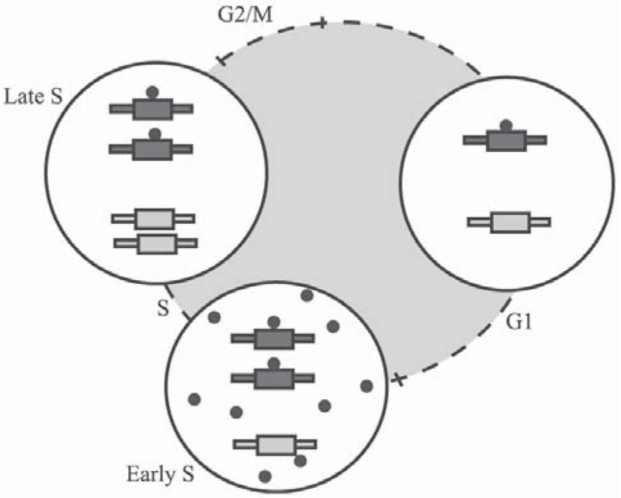


Fig. 4. Distinction of genes or alleles based on differential replication timing. The potential for differential timing of replication during S phase to maintain an epigenetic distinction between different genes or between different alleles of the same gene is depicted in this figure. The model is based on the concept that protein-DNA interactions become disrupted during DNA replication. These interactions must be re-established following replication. If certain proteins are present in limiting quantities, those genes or alleles that replicate earliest during S phase will gain preferential access to bind these proteins. In this way one gene or allele will bind a disproportionate quantity of a specific protein(s) and become distinguished from another gene or allele, even if both genes or alleles share similar protein-binding sequences. (Reproduced with permission from Simon, et al. [1999]).

repressed state of a gene is typically characterized by a condensed chromatin structure in which histones are deacetylated, DNA is hypermethylated and complexed with methylated DNA-binding proteins, and the gene is disassociated from the nuclear matrix and replicates relatively late during S phase. In the scheme depicted in Fig. 5, an initial tissue- and gene-specific demethylation event leads to a loss of binding of methylated DNA-binding proteins and their associated histone deacyetylase activity (*see* Fig. 5A,B). The histones then become acetylated and this facilitates a transition from the condensed, 30-nm chromatin structure to the decondensed, 10 nm structure, a process that has been termed *gene potentiation* (*see* Fig. 5B,C). Once the gene is in this open configuration, chromatin remodeling complexes can rearrange or displace nucleosomes to provide direct access for interaction between transcription factors and their cognate binding sites in the promoter region (*see* Fig. 5C–E). The bound transcription factors then attract the RNA polymerase complex to the proper transcriptional start site to initiate synthesis of RNA (*see* Fig. 5E–G).

It is possible that the initial demethylation event also facilitates, and/or is coincident with, enhanced association of the gene with the nuclear matrix. This could, in turn, provide enhanced proximity to replication machinery so that the gene would be subsequently replicated earlier during S phase. This could also provide more direct access to histone acetylases, chromatin remodeling complexes, transcription factors and RNA polymerase that may be embedded in, or associated with the matrix.

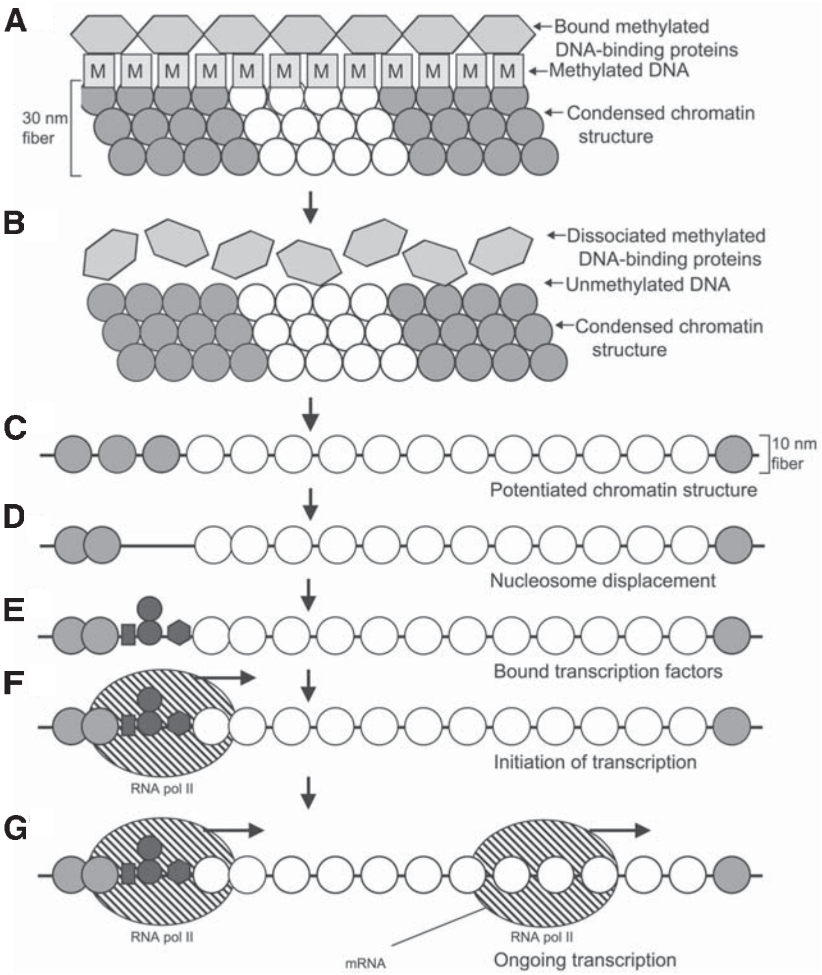

A
Bound methylated DNA-binding proteins
M M M M M M M M M M M M
Methylated DNA
30 nm fiber
Condensed chromatin structure
B
Dissociated methylated DNA-binding proteins
Unmethylated DNA
Condensed chromatin structure
C
10 nm fiber
Potentiated chromatin structure
D
Nucleosome displacement
E
Bound transcription factors
F
Initiation of transcription
RNA pol II
G
RNA pol II
mRNA
RNA pol II
Ongoing transcription

Fig. 5. Interactions among epigenetic mechanisms to regulate gene expression. Multiple epigenetic mechanisms contribute to regulation of transcriptional activity in mammals. An example of how this may occur is presented in this figure. **(A)** A fully repressed gene is often found to be hypermethylated, complexed with methylated-DNA binding proteins, comprised of deacetylated histones, and in a condensed chromatin structure characterized by a 30-nm structure that inhibits binding of transcription factors or RNA polymerase. **(B)** Derepression leading to transcriptional activation begins with demethylation of the gene, which in turn leads to dissociation of methylated-DNA binding proteins and acetylation of histones. **(C)** Potentiation of chromatin structure is marked by decondensation of the chromatin fiber from the 30 nm structure to the 10 nm *beads on a string* structure. **(D)** Displacement of nucleosomes creates an assayable DNase I hypersensitive site which marks the presence of naked DNA that is available for binding by transcription factors. **(E)** Binding of ubiquitous transcription factors to the core promoter region and tissue-specific transcription factors to enhancer regions attract the RNA polymerase II complex to the gene promoter. **(F)** Binding of the RNA polymerase II complex initiates transcription. **(G)** Ongoing transcription is characterized by sequential binding of multiple RNA polymerase II complexes to facilitate synthesis of multiple RNA transcripts. Open circles, nucleosomes within the transcribed portion of the gene; filled circles, nucleosomes between genes; squares containing Ms, DNA methylation; hexagons, methylated-DNA binding proteins; small filled hexagons, circles and rectangles, bound transcription factors; large, hatched oval, RNA polymerase II complex.

Epigenetics and Cloning

The critical role of epigenetic mechanisms governing gene regulation for normal mammalian development has recently come under particular scrutiny with the application of cloning of individuals by nuclear transplantation. In species that reproduce sexually, the union of male and female gametes at fertilization is the initial step in the development of a new offspring. Fusion of the haploid gametic genomes forms the diploid zygotic nucleus from which the genetic information required to direct the development of all subsequent cells of the embryo and fetus is derived. In both male and female mammals, gametogenesis is a complex process that includes epigenetic reprogramming processes that are unique to germ cells, and appear to contribute to preparing the gametic genomes to direct development of the ensuing embryo. When cloned mice are produced by transplantation of a somatic cell nucleus into an enucleated egg, the resulting zygotic nucleus is diploid, but it consists of two haploid genomes that have not undergone the reprogramming processes unique to gametogenesis. Presumably, reprogramming queues emanating from the ooplasm of the enulceated oocyte into which the somatic donor nucleus is transplanted are able to signal sufficient, rapid epigenetic reprogramming in the newly formed zygote.

Only a small proportion of embryos produced by *nuclear transplantation* (NT) develop to birth, and only a subset of these grow into fertile adults. Recent evidence has indicated that certain processes including X-chromosome inactivation and adjustment of telomere length appear to become properly reset in cloned mice. However, other epigenetic programs such as those affecting methylation patterns and expression of imprinted genes are often not properly reset and result in aberrant gene expression. Thus, proper epigenetic programming, which is typically manifest during gametogenesis, is indispensably required for normal mammalian development.

Interestingly, epigenetic reprogramming appears incomplete and/or unstable in cloned mice. If these mice are able to subsequently participate in natural breeding they produce offspring that appear relatively normal. Thus it appears that genetic and epigenetic programming is largely restored in the gametes of cloned mice by mechanisms that function uniquely in the germline. Inasmuch as the oocyte is a product of gametogenesis in the female, it is not surprising that it contains signaling factors that can potentially contribute to reprogramming of a transplanted somatic cell nucleus. However, those epigenetic mechanisms that normally function uniquely during spermatogenesis are not represented in a zygote produced by NT, and this deficiency may contribute to the very low success rate of full-term development of clones produced by NT.

Conclusions

As the human genome project nears its completion, the estimated number of genes in the mammalian genome has been reduced from an original concept of approx 100,000 genes to a revised estimate of approx 35,000 genes. However, even with this reduction, the challenge to each mammalian cell of correctly orchestrating the expression of these genes to facilitate the proper development and function of numerous different cell types remains a daunting task. It is therefore not surprising that mammals and other eukaryotes have evolved multiple mechanisms of gene regulation and that these mechanisms afford the cell multiple levels of control over gene expression. When considering the process of differential gene expression and the mechanisms which regulate this, it is important to bear in mind that there are various levels of control and that these are based on a combination of genetic and epigenetic mechanisms. A great deal of emphasis has been placed on the genetic mechanism of protein-DNA interactions between transcription factors and promoter binding sites for regulating gene expression. Although this mechanism is indeed critical to transcriptional regulation, it can only function if it is preceded by the proper functioning of the epigenetic mechanisms described in this chapter.

Glossary and Abbreviations

General

Cloning by Nuclear Transplantation (NT) The process of producing a genetically identical individual by transferring a diploid somatic nucleus into an enucleated egg.

Epigenetics The study of mitotically and/or meiotically heritable changes in gene function that cannot be explained by changes in primary DNA sequence.

Epigenetic Reprogramming The process by which epigenetic modifications are reversed or modified, especially during reproduction.

Gametogenesis The process by which germ cells differentiate to form the mature gametes, spermatozoa in males and ova in females.

Gene Expression Transcription and/or translation of a gene or gene product.

Genome One haploid complement of chromosomes.

Genomic Imprinting An epigenetic distinction of alleles in a parent-of-origin specific manner.

Germ Cells Those cells that give rise to the gametes and from which the genetic contribution to the subsequent generation is derived.

Housekeeping Gene A gene that is expressed in all or a significant majority of tissues or cell types.

Monoallelic Expression Expression of only one copy of a gene in a cell carrying two or more copies of the gene.

Replication Timing The time during S phase of the cell cycle at which a particular gene is replicated.

Semi-Conservative Replication of DNA The mechanism by which double-stranded DNA is replicated such that each new double helix includes one previously existing strand and one newly synthesized strand.

Tissue-Specific Gene A gene that is expressed in only a single tissue or cell type, or in only a subset of tissues or cell types.

X-Chromosome Inactivation Transcriptional silencing of an entire X chromosome used to compensate for differences in dosage of X-linked genes in XX and XY individuals.

Gene Expression

Coordinate Regulation The coordinated regulation of expression of more than one gene simultaneously.

Ectopic Gene Expression Expression of a tissue-specific gene in a tissue or cell type in which it is not normally expressed.

Promoter The region of a gene that regulates initiation of transcription.

Protein-DNA Interactions Binding of specific proteins (e.g. transcription factors) to specific sites in DNA.

RNA Polymerase The enzyme complex responsible for transcribing DNA to produce RNA.

Transcriptional Activation Initiation of transcription of a regulated gene.

Transcription Factors Proteins that bind to specific sequences in the gene promoter and regulate initiation of transcription.

DNA Methylation

CpG Island A region in mammalian DNA in which the representation of CpG dinucleotides is significantly greater than that found in the genome as a whole. Specifically, a region of > 100 base pairs in which the GC content is > 50% and the occurrence of CpG dinucleotides is > 60% of that predicted by random statistical occurrence.

De Novo Methylase An enzyme capable of adding methyl groups to both cytosines in a previously unmethylated CpG dinucleotide site in double-stranded DNA.

Demethylase Activity The active loss of methylation from both cytosines in a previously fully methylated CpG dinucleotide site in double-stranded DNA.

DNA Methylation A covalent modification of DNA produced by the addition of a methyl group to a cytosine base in mammalian DNA.

Fully Methylated DNA The presence of a methyl group on both cytosines in a CpG dinucleotide site in double-stranded DNA.

Hemimethylated DNA The presence of a methyl group on only one cytosines in a CpG dinucleotide site in double-stranded DNA.

Maintenance Methylase An enzyme capable of adding a methyl group to a cytosine in a hemimethylated CpG dinucleotide site in double-stranded DNA.

Methylated-DNA Binding Protein A protein that binds to DNA on the basis of the presence of methyl groups rather than on the basis of a specific base sequence.

Unmethylated DNA The absence of a methyl group on either cytosine in a CpG dinucleotide site in double-stranded DNA.

Chromatin

'Beads On A String' Structure The structure produced by the periodic presence of nucleosomes separated by a stretch of naked DNA in chromatin.

Boundary Elements Specific sequences in DNA that function to delimit DNA loop domains and/or to insulate one domain of DNA from a neighboring domain.

Chromatin DNA plus associated proteins.

Chromatin Composition The chemical status of chromatin that can be modified to include or exclude specific proteins or specific modifications to proteins (e.g. phosphorylation, acetylation, methylation).

Chromatin Structure The configuration of DNA plus associated proteins that can be relatively condensed to preclude transcription or relatively decondensed to facilitate transcription.

DNA or Chromosome Loop Domains Organization of genomic DNA into domains anchored at each end to the nuclear matrix.

Hyperacetylated Histones Histones, especially H4, that are predominantly acetylated.

Hypoacetylated Histones Histones that are predominantly unacetylated.

Histone Acetlyl Transferases (HATs) Enzymes responsible for acetylating histones.

Histone Deacetylases (HDACs) Enzymes responsible for deacetylating histones.

Matrix Attachment Regions (MARs) or Scaffold Attachment Regions (SARs) Specific sequences in DNA that bind to the nuclear matrix or scaffold either constitutively or facultatively.

Naked DNA DNA devoid of any associated proteins.

Nuclear Matrix A three-dimensional proteinaceous structure within the nucleus (aka the nuclear scaffold).

Nucleosomes Histone octamers found in chromatin.

Open or Potentiated Chromatin Chromatin that is decondensed such that it is accessible for transcription by RNA polymerase.

Suggested Readings

Introduction

Russo, E., Martienssen, R., and Riggs, A.D. (eds.) (1996) Epigenetic Mechanisms of Gene Regulation, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

DNA Methylation

Ariel, M., Cedar, H., and McCarrey, J. R. (1994) Developmental changes in *Pgk-2* gene methylation during spermatogenesis in the mouse: Reprogramming occurs in epididymal spermatozoa, Nat. Genet. 7, 59–63.

Ben-Porath, I. and Cedar, H. (2000) Imprinting: focusing on the center, Curr. Op. Genet. Dev. 10, 550–554.

Bestor, T. H. (2000) The DNA methyltransferases of mammals, Hum. Mol. Genet. 9, 2395–2402.

Brandeis, M., Kafri, T., Ariel, M., Chaillet, J. R., McCarrey, J. R., Razin, A., and Cedar, H. (1993) The ontogeny of allele-specific methylation associated with imprinted genes in the mouse, EMBO J. 12, 3669–3677.

Gardiner-Garden, M. and Frommer, M. (1987) CpG islands in vertebrate genomes, J. Mol. Biol. 196, 261–282.

Hendrich, B. and Bird, A. (1998) Identification and characterization of a family of mammalian methyl-CpG binding proteins, Mol. Cell. Biol. 18, 6538–6547.

Jones, P. L., Veenstra, G. J., Wade, P. A., Vermaak, D., Kass, S. U., Landsberger, N., Strouboulis, J., and Wolffe, A. P. (1998) Methylated DNA and MeCP2 recruit histone deacetylase to repress transcription, Nat. Genet. 19, 187–189.

Kafri, T., Ariel, M., Shemer, R., Urven, L., McCarrey, J. R., Cedar, H., and Razin, A. (1992) Dynamics of gene specific DNA methylation in the mouse embryo and germ line, Genes Dev. 6, 705–714.

Nan, X., Tate, P., Li, E., and Bird, A. P. (1996) DNA methylation specifies chromosomal localization of MeCP2, Mol. Cell. Biol. 16, 414–421.

Ng, H-.H. and Bird, A. (1999) DNA methylation and chromatin modification, Curr. Op. Genet. Dev. 9, 158–163.

Ramsahoye, B. H., Biniszkiewicz, D., Lyko, F., Clarck, V., Bird, A. P., and Jaenisch, R. (2000) Non-CpG methylation is prevalent in embryonic stem cells and may be mediated by DNA methyltransferase 3a., Proc. Natl. Acad. Sci. USA 97, 5237–5242.

Riggs, A. D. and Porter, T. N. (1996) X-chromosome inactivation and epigenetic mechanisms, in: Epigenetic Mechanisms of Gene Regulation, (Russo, E., Martienssen, R., and Riggs, A.D., eds.) Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, pp. 231–248.

Shemer, R. and Razin, A. (1996) Establishment of imprinted methylation patterns during development, in: Epigenetic Mechanisms of Gene Regulation, (Russo, E., Martienssen, R., and Riggs, A.D., eds.) Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, pp. 215–230.

Tilghman, S. M. (1999) The sins of the fathers and mothers: genomic imprinting in mammalian development, Cell 96, 185–193.

Zhang, L. P., Stroud, J. C., Walter, C. A., Adrian, G. S., and McCarrey, J. R. (1998) A gene-specific promoter in transgenic mice directs testis-specific demethylation prior to transcriptional activation *in vivo*, Biol. Reprod. 59, 284–292.

Chromatin Structure and Composition

Carey, M. and Smale, S. T. (2000) Transcriptional regulation in eukaryotes, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, p. 640

Kramer, J. A., McCarrey, J. R., Djakiew, D., and Krawetz, S. A. (1998) Differentiation: The selective potentiation of chromatin domains, Development 9, 4749–4755.

Urnov, F. D. and Wolffe, A. P. (2001) Chromatin remodeling and transcriptional activation: the cast (in order of appearance), Oncogene 20, 2991–3006.

Varga-Weisz, P. D. and Becker, P. B. (1998) Chromatin-remodeling factors: machines that regulate? Curr. Op. Cell. Biol. 10, 346–353.

Wolffe, A. P. (1992) Chromatin, Academic Press, San Diego, CA, p. 213.

DNA Loop Domains and Association with the Nuclear Matrix

Bell, A. C., West, A. G., and Felsenfeld, G. (2001) Insulators and boundaries: versatile regulatory elements in the eukaryotic genome, Science 291, 447–450.

Klaus, A., McCarrey, J. R., Farkas, A., and Ward, W. S. (2001) Changes in DNA loop domain structure during spermatogenesis and embryogenesis, Biology of Reproduction 64, 1297–1306.

Mirkovitch, J., Mirault, M. E., and Laemmli, U. (1984) Organization of the higher-order chromatin loop: specific DNA attachment sites on nuclear scaffold, Cell 39, 223–232.

Sotolongo, B. and Ward, W. S. (2000) DNA loop domain organization: the three-dimensional genomic code, J. Cell. Biochem. Suppl. 35, 23–26.

Stein, G. S. and Berezney, R. (2000) 25 years of contributions to characterizing gene expression and replication within the three-dimensional context of nuclear architecture, Crit. Rev. Eukaryot. Gene. Expr. 10, v–vi.

Replication Timing

Goldman, M. A., Holmquist, G. P., Gray, M. C., Caston, L. A., and Nag, A. (1984) Replication timing of genes and middle repetitive sequences, Science 224, 686–692.

Hansen, R. S., Canfield, T. K., Lamb, M. M., Gartler, S. M., and Laird, C. D. (1993) Association of fragile X syndrome with delayed replication of the FMR1 gene, Cell 73, 1403–1409.

Kitsberg, D., Selig, S., Brandeis, M., Simon, I., Keshet, I., Driscoll, D. J., Nichols, R. D., and Cedar, H. (1993) Allele-specific replication timing of imprinted gene regions, Nature 364, 459–463.

Simon, I., Tanzen, T., Rubinoff, B., McCarrey, J., and Cedar, H. (1999) Asynchronous replication of imprinted genes is established in the gametes and maintained during development, Nature 401, 929–932.

Epigenetics and Cloning

Eggan, K., Akutsu, H., Hochedlinger, K., Rideout, III, W., Yanagimachi, R., and Jaenisch, R. (2000) X-chromosome inactivation in cloned mouse embryos, Science 290, 1578–1581.

Humpherys, D., Eggan, K., Akutsu, H., Hochedlinger, K., Rideout, 3rd, W. M., Biniszkiewicz, D., et al. (2001) Epigenetic instability in ES cells and cloned mice, Science 293, 95–97.

Krawetz, S. A., Kramer, J. A., and McCarrey, J. R. (1999) Reprogramming the male gamete genome: a window to successful gene therapy, Gene 234, 1–9.

Lanza, R. P., Cibelli, J. B., Blackwell, C., Cristofalo, V. J., Francis, M. K., Baerlocher, G. M., et al. (2000) Extension of cell life-span and telomere length in animals cloned from senescent somatic cells. Science. 28, 665–669.

Rideout, III, W. M., Eggan, K., and Jaenisch, R. (2001) Nuclear cloning and epigenetic reprogramming of the genome, Science 293, 1093–1098.

Wakayama, T., Perry, A. C. F., Zuccotti, M., Johnson, K. R., and Yanagimachi, R. (1998) Full-term development of mice from enucleated oocytes injected with cumulus cell nuclei, Nature 394, 369–374.

7 Gene Families and Evolution

Ben F. Koop

Introduction

Gene families refer to two or more genes that come from a common ancestral gene in which the individual members of the gene family may or may not have a similar function. The idea of gene families implicitly invokes a process in which an original gene exists, is duplicated and the resulting gene products evolve. The most common result of gene duplication is that mutation renders one of the products nonfunctional and in the absence of conserving natural selection, one of the members becomes no longer recognizable. Gene families may be clustered or dispersed and may exchange with each other through the mechanisms of gene conversion or unequal crossover. Therefore understanding the processes of molecular evolution are essential to understanding what gene families are, where they came from and what their function might be. In a sense, duplicate genes allow for more evolutionary potential. At first glance this could be beneficial; if one gene incurred a lethal mutation the other gene simply takes over, there is some protection from mutation based on redundancy. Having two identical genes could result in twice as much product; this may or may not be beneficial in a cell where the integration of thousands of gene products must be coordinated and slight concentration differences can alter biochemical pathways.

Antiquity of Gene Families

Almost all genes belong to gene families. Evidence from sequence or structural similarity indicates that all or at least large parts of genes came from ancestral genes. Analysis of the human genome has shown that over half of the human genome is comprised of clearly identifiable repeated sequences. Although much of this is owing to self-replicating transposons, i.e., mobile genetic elements, over 5 % of the genome has been involved in large segmental duplications in the past 30 million years. If we look further into the past using evidence from protein similarity of three or more genes that occur in close proximity on two different chromosomes, we find over 10,310 gene pairs in 1,077 duplicated blocks contain 3,522 distinct genes. Because our observations are based on genes that retain similarity, only a small fraction of the ancient duplications can be detected by current means. What is clearly evident is that a very large part of the human genome has come from duplications and that duplication is a very frequent event.

With the evidence showing that gene duplication plays a major role in modern genomes, the question is where did it all begin? How many genes did life start with? Various estimates of the minimal gene set suggest that as few as 250 genes could provide the minimal number of components necessary to sustain independent life. The number of genes in mycoplasms ranges from 500–1500 genes and in bacteria from 1000–4000 genes. Yeast (*S. cervisiae*) has ~ 6000 genes, worms (*C. elegans*) have ~ 18,000 genes, the fruit fly (*D. melangaster*) has ~13,000 genes, a plant, *Arabidopsis*, has 26,000 genes, and humans have at least 30,000 genes. The difference between 250 or 1000 genes and 30,000–40,000 genes is only two orders of magnitude yet the difference in the complexity of life forms seems far greater than could be explained by a simple gene count. Certainly with the increased number of genes comes the opportunity for more complexity in terms of gene interaction. But can synergy alone explain the differences in morphological complexity? Partial explanations invoke more complex differential splicing to account for a higher proportional number of protein products stemming from only 30,000 genes, but one wonders if this observation is merely an artifact of the few numbers of whole genomes available to us.

Origins

Gene duplications can come from a variety of sources including whole or partial genome duplication. Polyploidy results from a failure of chromosome segregation during the cell division of gametes. The most distinguishing feature of polyploidy is that it effects all of the genes simultaneously so that the relative proportion of genes within cells remains the same. Among plants and invertebrates, polyploidy is quite common and in many species it has little effect on phenotype. Ohno has argued that whole genome duplications are the most important events in evolution yet others have suggested that polyploidy has no effect on phenotype. More recent discussions acknowledge the potential that polypoidy brings to gene family evolution but also appreciate the role of complexity of gene interactions in determining the impact of whole genome duplications.

In vertebrates, polyploidy is quite rare. Most of the 188 examples of genome duplication have been found in amphibians, reptiles and some fish (salmon). In these instances, polyploid species have undergone dramatic changes to reestablish diploidy through chromosome loss, mutation, and rearrangement. Tetraploid genomes have no trouble going through cell division as long as chromosomes remain very similar. But as mutations arise and duplicated chromosomes begin to differ, cell division can no longer insure equal division of genetic material to germ cells and severe imbalances can occur during chromosomal segregation. The initial transition phase from tetraploid to diploid results in huge losses of gametes and developing young. In salmon it is estimated that approx 50 million years after a polypoid change, only 53% of duplicate genes remain. The result of duplication by whole or partial genomes can result in large changes in gene number but there are major difficulties in cell replication that must be overcome.

The rapid increase in genome size through polyploid events has been used to explain the increased size of mammalian genomes. Ohno suggested that two rounds of genome duplication occurred early in vertebrate history. This may explain the Cambrian explosion in which vertebrates appeared in paleontological records quite rapidly. Evidence for two rounds of genome duplication comes from vertebrates having four times the number of developmental regulator genes (*Hox*, *Cdx*, *MyoD*, *60A*, *Notch*, *elav*, *btd/SP...*) as *Drosophila*. While this concept has become very popular in the literature, recent studies examining expected phylogenetic relationships among genes have called into question whether the number of genes was a result of two genome-wide duplications or simply a result of ongoing, frequent genome segment duplications. While the primary support for quadruplicated genomes comes primarily from chromosomes 2, 7, 12, and 17, which contain the *Hox* gene clusters, a more extensive examination of the number of homologous genes within humans as compared to genes within *Drosophila* was unable to resolve the question of whether two whole genome duplications gave rise to modern vertebrate genomes.

Mechanisms

Duplication of large blocks of DNA can not be explained by chromosomal segregation errors. Mechanisms of large segmental duplication are varied and the role of transposable elements in gene duplication is often cited as a primary cause. Transposable elements and in particular retrotransposable elements are highly repetitive dispersed sequences that can replicate independent of nuclear division. In humans they comprise over 45% of the genome. While there are a number of instances where retroposons have been found at the junctions of duplicated segments, there are also a number instances where they have not. To understand why gene segment duplications appear common, it is perhaps important to look at the DNA molecule itself. DNA is composed of duplex strands held together by hydrogen bonds whose strength varies. In a fluid environment, various local salt concentrations, temperatures, physical torsion forces and local nucleotide compositions (e.g., levels of G + C, simple repeats) can result in temporary separation of strands of duplex DNA. If similar sequences are found in the same physical location, unstable heterologous duplexes can form. Heterologous pairing or single-strand conditions are prone to stress and breakage. These situations are repaired correctly in the vast majority of cases but occasionally mistakes are made that result in new gene neighbors. The possibility of error is particularly high during cell division when DNA is being replicated and when similar sequences are in close proximity. The potential impact of highly repeated transposable elements as a destabilizing factor and a potential focal point for rearrangement becomes clear in our genome. It is therefore somewhat surprising to find that many of the duplications are not flanked by repeat elements. What is clear is that duplication involves local chromosome instability that results in breakage and aberrant repair of the ends.

Factors that promote segmental duplication include close proximity and high sequence similarity. It then follows that duplications resulting in adjacent genes would be more susceptible to further changes than duplications resulting in dispersed genes. Furthermore, adjacent duplications create far fewer chromosomal segregation problems during cell division and therefore should be found more often in the genome. What is observed is that large segmental duplications involving multiple genes are dispersed throughout the genome whereas duplications involving single gene segments are both dispersed and in close proximity. There are many instances of clustered gene families (globins, *Hox*, Ig, Tcr, Mhc, and rRNA).

Among tandemly duplicated gene segments there is the possibility of extensive gene conversion and unequal crossing over (*see* Fig. 1). The later is the predominant mode of change. Unequal crossing over between dispersed genes results in extreme difficulties in chromosomal segregation in cell division but gene conversion does not.

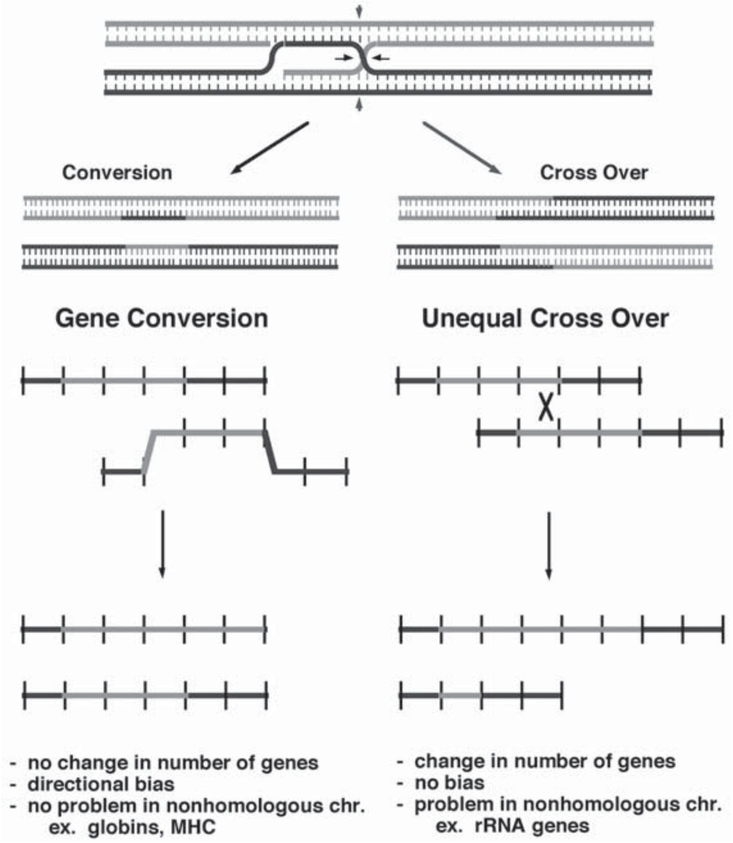

Conversion
Cross Over
Gene Conversion
Unequal Cross Over
X
- no change in number of genes
- directional bias
- no problem in nonhomologous chr.
ex. globins, MHC
- change in number of genes
- no bias
- problem in nonhomologous chr.
ex. rRNA genes

Fig. 1. Gene conversion and unequal crossing over mechanisms of communication among gene family members. The arrows indicate possible break points that would result in either conversion or crossover results.

Gene conversion replaces the sequence of one family member with the sequences of another close (>90% similarity) member but it does not effect the total number of genes. Gene conversion requires DNA strand breakage followed by strand migration to a similar gene and the formation of a heteroduplex. DNA repair mechanisms then repair differences in the heteroduplex often using one strand corresponding to the unaffected homologous chromosome as a template. The heteroduplex then resolves and may go back to its original location carrying with it DNA changes. Heteroduplex formation is often temporary and most often occurs between alleles of the same gene though occasionally it may affect paralogous genes in which more than 100 bases are greater than 95% similar. Depending on the resolution of the heteroduplex and biases in mismatch repair, adjacent base differences may both reflect one or the other parental strand, or they may reflect a combination of parental strands. The end result is that the total genetic variation is reduced but a particular gene may increase its number of alleles. Genetic variation at one gene can increase over a single conversion event, but over multiple conversion events variation is reduced.

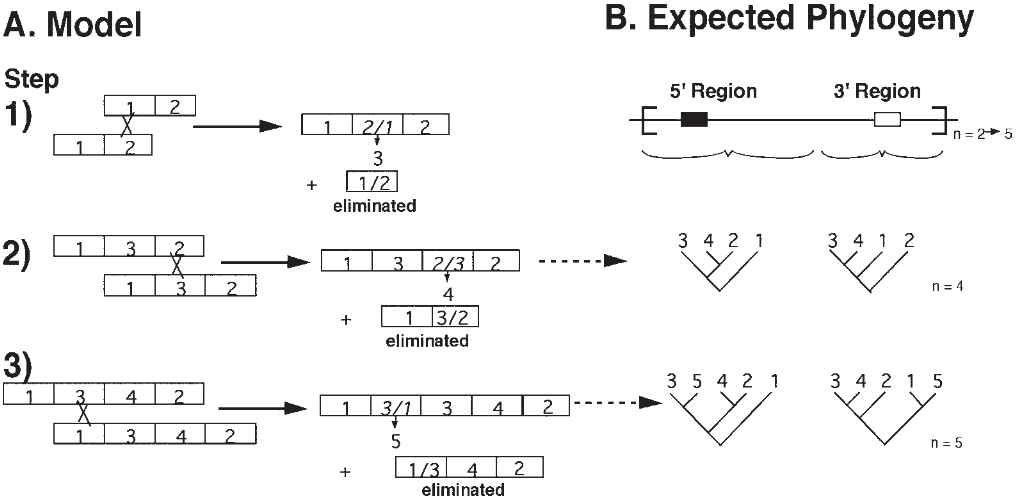


Fig. 2. Unequal crossing over between tandemly arrayed gene family members. This model **(A)** assumes a break point near the middle of the duplicated segments. The expected phylogeny **(B)** represent sequence relationships between the 5' and 3' regions of the duplicated gene segments.

The resolution of heterduplexes formed from the invasion of a DNA strand from one gene segment into the duplex of a similar, adjacent duplicate can also result in unequal crossing over. Unequal crossing over changes the number of genes. For example, in a tandem arrangement, unequal crossing over results in one chromosome with one duplicate and the other chromosome with three duplicates where the front and the back parts of the single duplicate and the middle duplicate of the triplicated segment reflect different origins. Figure 2, shows three successive unequal crossing-over events and shows the expected phylogenetic relationships of the front (5') and back (3') parts of the gene. It is evident from the final trees for the 5' and 3' parts of a tandemly arrayed gene segment that it is possible to describe some of the major, more recent evolutionary events that have occurred. Because information is lost due to the fixation of one of the cross over products in each population, it may never be possible to obtain a complete historical picture. But we can see in examples from the literature that unequal crossing over is the major factor in clustered gene families.

Variation

Genetic variation increases when the number of duplicates increases but it is decreased when the number of duplicates decreases. It is important to remember several tenets of unequal crossing over. First, the ultimate fate of duplicates undergoing multiple unequal crossing overs is to return to a single copy unless selection maintains multiple copies. Second, while the overall variation may increase over a single event, the result over multiple expansions and deletions is homogenization of duplicates (example rRNA genes). Third, unequal crossing over between dispersed gene segments often results in fatal problems in cell division. Lastly, unequal crossing over is the predominant mechanism that increases or decreases the number of gene family members in clusters. It appears that the factors that promote duplication include proximity, high similarity, larger numbers of existing duplicates and internal sequences that are prone to breakage. Given these factors, it is perhaps surprising that we do not see more evidence of repetitive elements playing a larger role in gene duplication. At the same time it becomes easy to see the complex evolution and interactions among both dispersed and clustered gene families.

Genes and Domains

Duplication can involve very large stretches of DNA, whole genes or even parts of genes. Of 1077 duplication blocks containing three or more genes in the human genome, 159 contained 3 genes, 137 contained 4 genes, and 781 contained five or more genes. At the same time we often see clusters of gene family members. This indicates that duplications often involve one gene or even parts of genes. Clearly the mechanisms of duplication outlined earlier play a major role at all levels of gene family evolution. However, it is important to remember that the events that are most evident are those that are fairly recent or those that involve conserved genes. Sequence similarity for older duplications of noncoding DNA rapidly fades. It is our focus on function that draws us to study genes. As mentioned, several genes within larger segments can be duplicated but perhaps just as interesting, parts of genes (introns and groups of introns) can be duplicated. This is particularly interesting because genes are composed of functional domains. Remarkably, there may be fewer than 1000 classifications in existence. Domains can be mixed, matched, duplicated and modified to provide novel functions within genes as well as between genes. Only 94 of the 1278 protein families in our genome appear to be specific to vertebrates. That may be an overestimate resulting from our inability to recognize similarity. It appears that the 30,000-plus genes in the human genome are not novel but simply products of duplications and mixing and matching of existing genes and domains to create new genes and new functions.

Species Evolution and Gene Evolution

The study of gene evolution is incomplete without the study of species evolution. Gene evolution and species evolution is not the same but knowledge of one greatly benefits our knowledge of the other. Modern species are the result of a dense network of transient species that occasionally give rise to other species. Using paleontological records as well as morphological, physiological, and developmental studies of extant and extinct life forms we are able to trace some of the origins of modern species. But numerous gaps in our understanding remain. Gene evolution can occur within a species but when a speciation event occurs, gene evolution within each new lineage is independent of gene evolution in other lineages. For example, gene evolution within humans is independent of gene evolution within chimpanzees. To illustrate this point, Fig. 3A shows a gene duplication in a common ancestor of species A and B followed by a speciation event and separate A and B lineages. Below Fig. 3A is the corresponding gene phylogenetic tree. Note that the timing of speciation events can be used to time major events in gene evolution. In Fig. 3A, gene duplication occurred before the speciation event. In Fig. 3B, a more complex gene evolution is shown. Within each species one of the genes has been eliminated and the other has been duplicated so that each species has two genes, which appear to have arisen after the speciation event.

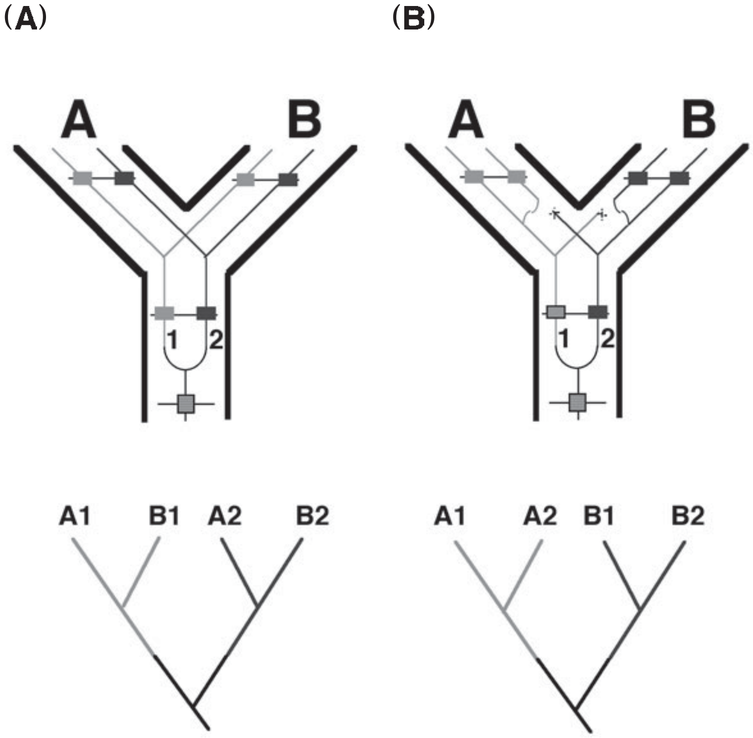


Fig. 3. Species evolution (in bold outline) and gene evolution (light lines). **(A)** Shows a gene duplication in a common ancestor of species A and B followed by a speciation event and separate A and B lineages. **(B)** A more complex gene evolution is shown. Within each species one of the genes has been eliminated and the other has been duplicated so that each species has two genes, which appear to have arisen after the speciation event.

While many other scenarios can occur, these two illustrations show how even in relatively simple cases, one must be cautious when interpreting gene trees. A number of studies have used these methods to identify new gene family members (Goodman et al., 1975). Slightom et al. (1987) was one of the first to use combined gene and species studies to examine genetic mechanisms of change. What is clear is that the use of both gene trees and species trees can be a very powerful method of studying species evolution and gene evolution.

Examples

The globin gene family offers one of the most studied and widely discussed examples of gene family evolution. Globin proteins function to transport oxygen and are found in bacterial, plant and animal kingdoms. In vertebrates, a series of gene duplications (*see* Fig. 4) correspond to major events in the evolution of man. A monomeric globin gene duplicated 600–800 million years ago to give rise to myoglobin (functional in muscle) and hemoglobin (functional in blood). 450–500 million years ago another duplication gave rise to an α form and a αβ form. About this same time hemoglobin changed from a monomeric form to a tetrameric form (2α subunits and 2β subunits),

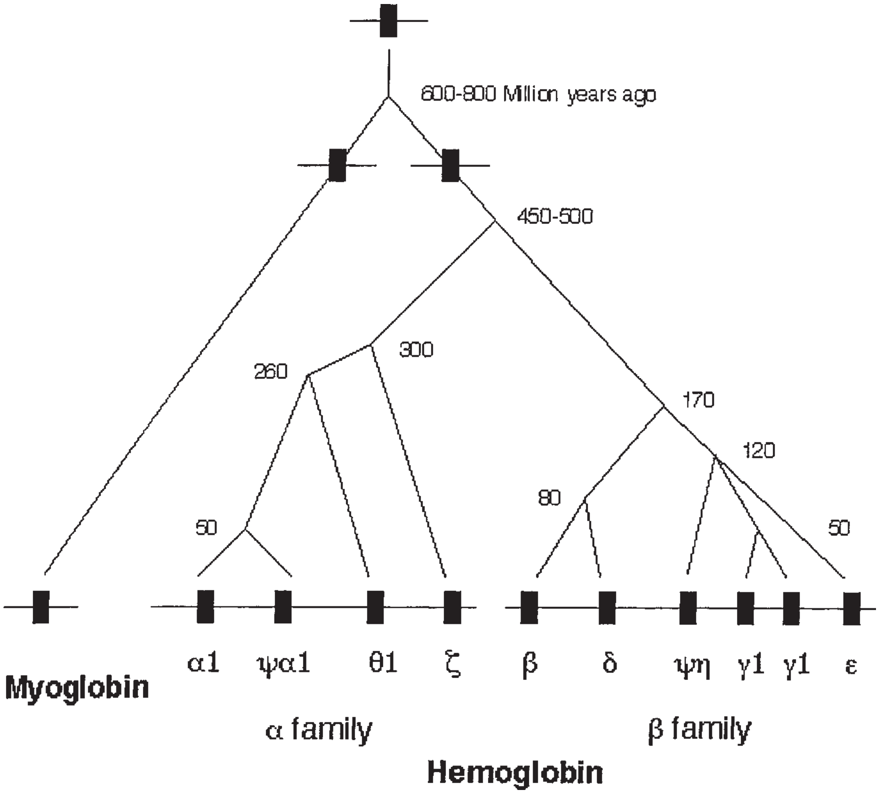


Fig. 4. Vertebrate globin evolution.

which Permitted oxygen transport in a much broader range of physiological conditions. The time frame for this event is supported by molecular clock estimates and by the fact that fish, amphibians, birds, and mammals all have a tetrameric hemoglobin with 2α and 2 β subunits. Over the next several hundred million years homeothermy evolved. Prior to the separation of mammals and marsupials (~150 million years ago), the β gene duplicated and gave rise to a form (ε) that is only found expressed in embryo's and a form (β) that is expressed in adults. Subsequent to the separation of marsupials and true mammals (eutherians), the embryonic form again duplicated into three separate genes. At this same time, major changes in the placenta made it possible to *hatch* eggs inside the body and allow for prolonged development prior to birth. In most eutherians there are also two adult forms (δ and β). In primates as well as cows (artiodactyls) the γ gene is found expressed in juveniles. In the α lineage, a similar chain of events occurred giving rise to embryonically expressed genes and genes expressed in the adult. What is clear from the study of globin gene evolution is that duplication allowed subsequent specialization, which in turn allowed for greater physiological complexity within species. The study of globin gene evolution within and between species also provided evidence for mechanisms of change which helps us to understand several globin-related diseases (thalasemia).

Over the past three decades, many other gene families have been studied and in each case these studies have provided important information about the numbers of gene family members, distribution, and functional specialization as well as information about the rates, modes, and mechanisms of change within each family. There is a tremendously diverse array of gene families with their own story. Some of the gene families such as the immunoglobulin super gene family incorporate evolutionary mechanisms of change into their function. In T-cell receptor genes, recombination, and alternative splicing of up to a hundred different gene family members results in the production of up to 10^{15} different receptors. This capability greatly facilitates the ability of T-cell receptors to recognize foreign proteins in the body and is a critical component in the overall function of the immune system. The rRNA genes also incorporate mechanisms of change into their overall functionality. In humans ~ 300 tandemly arrayed rRNA genes undergo extensive unequal crossing over to maintain several hundred nearly identical genes. In most cases, duplication of genes results in new and better control of physiology, growth, and development.

Significance

Evolutionary biologists have suggested that gene duplication followed by modification is the most important mechanism for generating new genes and biochemical processes. This has made it possible to evolve complex organisms from primitive ones. Based on traditional models, once duplications occur, one of the two genes is redundant and thus freed from functional constraints. All mutations, even missense or nonsense mutations, occurring in a redundant gene will be neutral unless by chance a mutation or combination of mutations results in a modified gene with some novel function. There are problems with this model because both gene products may still be subject to selection. Altered gene duplicates often result in products that can compete with each other for limited cis-acting promoter/enhancer molecules, or can produce altered products that can interfer with biochemical processes and molecular interactions. Furthermore there are examples (primates, opsins) where alternative alleles code for different functions and when gene duplication occurs, both functions are separated and free to be independently expressed (e.g., primate opsin genes resulting in bicolor versus tricolor vision). While there are a number of cases where bifunctionality precedes gene duplication, the duplication process itself is more dependent on factors such as the number of existing gene family members, proximity, similarity, and internal sequence fragile sites. Since duplications can involve many different genes or no genes at all, it is unlikely that bifunctionality precedes most duplications but it certainly affects the rate at which duplications survive and are fixed in populations. Duplication, rearrangement, and mutation of genes and domains are the critical forces in the evolution of gene families. The diverse affects of gene duplications offers a tremendous opportunity for diversification of gene function and is essential when evolving from a more generalized function to more specialized functions. Gene families have made it possible to evolve complex organisms from primitive organisms.

Suggested Readings

Baltimore, D. (2001) Our genome unveiled, Nature 409, 814–816.

Claverie, J.-M. (2001) What if there are only 30,000 human genes?, Science 291, 1255–1257.

Graur, D. and Li, W.-H. (2000) Fundamentals of Molecular Evolution, 2nd ed., Sinauer Associates, Inc., Sunderland, MA.

Goodman, M., Moore, G. W., Matsuda, G. (1975) Darwinian evolution in the genealogy of hemoglobin, Nature 253, 603–608.

Hughes, A. L. (1999) Adaptive Evolution of Genes and Genomes, Oxford University Press, New York, NY.

International Human Genome Sequencing Consortium (2001) Initial sequencing and analysis of the human genome, Nature 409, 890–921.

Lewin, B. (2000) Genes VII, Oxford University Press, New York, NY.

Otto, S. P., Whitton, J. (2000) Polyploid incidence and evolution, Annu. Rev. Genet. 34, 401–437.

Sankoff, D. and Nadeau, J. H. (eds.) (2000) Comparative Genomics, Kluwer Academic Press, Dordrecht, The Netherlands.

Slightom, J. L., Theissen, T., Koop, B. F., and Goodman, M. (1987) Orangutan fetal globin genes: nucleotide sequences reveal multiple gene conversions during hominid phylogeny, J. Biol. Chem. 262, 7472–7483.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J, Sutton, G. G., et. al. (2001) The sequence of the human genome, Science 291, 1304–1351.

8 Repetitive DNA
Detection, Annotation, and Analysis

Jerzy Jurka

Introduction

Eukaryotic genomes are composed primarily of nonprotein-coding DNA. The most actively studied portion of this DNA is called *repetitive DNA*, which is produced in multiple copies by a variety of mechanisms. Repetitive DNA represents the most recent addition to nonprotein coding DNA and is expected to hold important clues to the origin and evolution of genomic DNA. There are good reasons to believe that contemporary mechanisms underlying the origin and evolution of repetitive DNA are essentially the same as mechanisms that generated other nonprotein-coding sequences in the distant past.

Repetitive DNA began surfacing in unprecedented detail as soon as critical mass of human sequence data permitted comparative analyses. This set the stage for a new era of repeat studies dominated by computer-assisted sequence comparisons. Currently, 42% of the human genome is recognizable as being derived from repetitive DNA. This proportion may vary from species to species in a seemingly arbitrary manner and the exact reasons why some eukaryotic species preserve more DNA than others are not well understood.

Studies of repetitive DNA are important not only per se, but also in the context of genome biology, including its structure, stability, and evolution. Repeats often obscure proteins and other regions of biological significance and for this reason they need to be identified and filtered out of the sequence data to facilitate such studies. Identification of repeats is also necessary for probe and primer design in DNA-DNA hybridization and polymerase chain reaction (PCR) studies, respectively. Inevitably, they are increasingly being studied in various biological contexts including but not limited to phylogenetic analysis, population studies, gene polymorphism, and chromosomal organization.

Simple Sequence Repeats (SSRs)

There are two basic classes of repetitive DNA sequences: 1) those expanded spontaneously on-site; and 2) those transposed from somewhere else as copies of transposable elements (TEs). These two classes are not totally independent because TEs can initiate or stimulate on-site expansion of repetitive DNA. The most common repeats generated on site are tandem repeats, often referred to as simple sequence repeats or

SSRs. Typically, tandem repeats with a unit size of 10 bp or less are referred to as *microsatellites*. Tandem repeats with a unit size over 10 bp are called *minisatellites*. There is a significant *twilight zone* between micro- and minisatellites, usually applicable to repeats with unit size 7–14 bp, which is listed in either category in the scientific literature.

The number of units, i.e., overall length of micro- and minisatellites, can vary from generation to generation and this property makes them very useful in studies of sequence polymorphism in eukaryotic populations. Growing evidence indicates that micro- and minisatellite expansion occurs by different mechanisms: the former mostly due to *polymerase slippage*, and the latter due to an illegitimate recombination process stimulated by double-stranded breaks. Tandem repeats are often transformed to a cryptically simple DNA composed of various sequence motifs rearranged and often obscured by mutations. Tandemly repeated sequences include satellite DNA. Satellites are primarily located in centromeres, whereas other tandem repeats tend to be interspersed within genomic DNA. Like other tandem repeats, satellites are quite variable and even closely related species may carry completely unrelated satellites. Satellite variability may be fueled by mechanisms similar to those involved in minisatellite variability.

Transposable Elements (TEs)

The major source of *interspersed repetitive DNA* are transposable elements (TEs). There are two major classes of TEs in the eukaryotic organisms: class 1, retro-elements; class 2, DNA transposons; including *rolling-circle* transposons, which was recently discovered in plants and nematodes (*see* Fig. 1).

Retroelements include long interspersed nuclear elements (LINEs) and elements related to retroviruses, including some *domesticated* endogenous retroviruses. All retroelements use reverse transcriptase to copy their RNA to DNA as a part of their reproduction process. LINEs generate a variety of retropseudogenes including SINE elements.

In the case of mammalian LINE1 (L1) element, reverse transcription is initiated (primed) by a reverse trancriptase-generated nick in host DNA. A second nick is generated on the opposite strand leading to target site duplication (TSD) where the duplicated target is represented by a short, ~15 bp long DNA fragment delimited by the nicks. The final integration is probably completed by the host replication system. Unlike LINEs, retroviruses appear to be inserted in a separate step after they are reverse transcribed to DNA. There is no specific mechanism for excision of retroelements although integrated retroviruses can be deleted due to homologous recombination between long terminal repeats (LTRs), leaving behind a single (solo) LTR repeat.

DNA transposons (class 2) encode transposase, which is involved in insertion and excision of these elements to and from host DNA. The transposase recognizes terminal inverted repeats (TIRs). Replication of a DNA transposon is accomplished by the host replication system. If excision does not occur, the transposon becomes permanently integrated usually as an inactive repetitive element.

The third class of eukaryotic TEs is represented by complex rolling-circle (RC) transposons. In addition to a cleavage and replication transposase, RC transposons use enzymes such as helicases and the single-strand DNA-binding protein, probably adopted from the host. RC transposons integrate at AT dinucleotides without target site duplication.

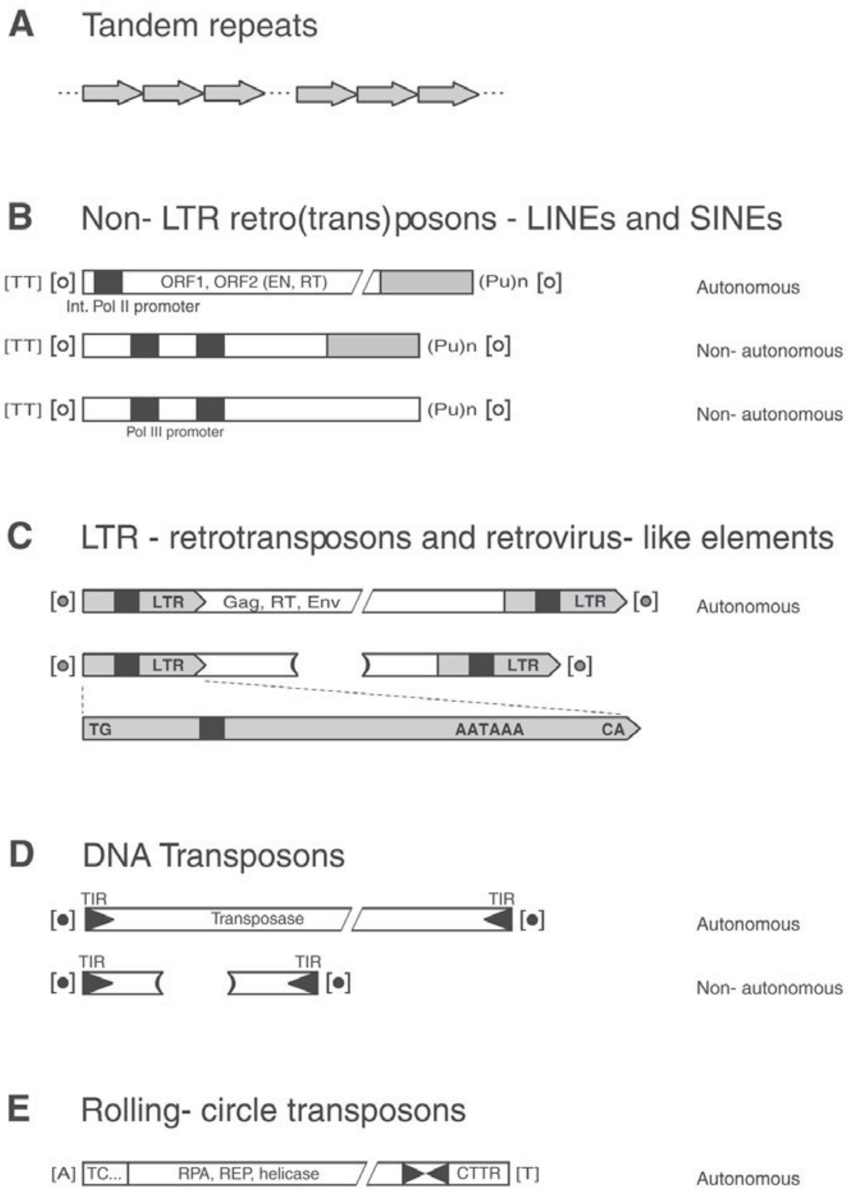


Fig. 1. Basic categories and biological characteristics of repetitive elements. **(A)** Tandem repeats including minisatellites, microsatellites, and satellites. **(B)** Structure of LINE (autonomous) and SINE (nonautonomous) retroelements; black boxes show transcription promoters and (Pu)n indicate purine (A or G) tails. Target site duplications (TSDs) and other target components throughout the figure are indicated by brackets [ ]. **(C)** LTR-retrotransposons and retrovirus-like elements. Characteristic sequence features of LTRs: 5′ TG, 3′ CA, and polyadenylation signal AATAAA are indicated in the enlarged long terminal repeat. **(D)** Autonomous and nonautonomous DNA transposons. Black triangles at both ends indicate terminal inverted repeats (TIRs). **(E)** Autonomous and nonautonomous rolling-circle transposons. Characteristic 5′ TC, 3′ CTTR and hairpin-like structures (inverted black triangles) are indicated.

All classes of autonomous TEs in eukaryotes are associated with nonautonomous elements that do not encode any active enzymes. They depend on their autonomus relatives for reproduction and insertion into the genome. In this context, autonomous TEs can play a role of *mutator genes* that must be restricted or tightly controlled by the host. In general, only few active TEs at a time appear to find favorable circumstances for proliferation in any given population. They produce a discrete *genomic fossil record* of repetitive families/subfamilies derived from a limited number of actively expressed source genes or active TEs. Both autonomous and nonautonomous elements have their actively expressed source genes. Source genes can be active for millions of years but are eventually replaced by their variants or become extinct. Interestingly, copies of nonautonomous elements, particularly short ones, tend to predominate over the autonomous ones.

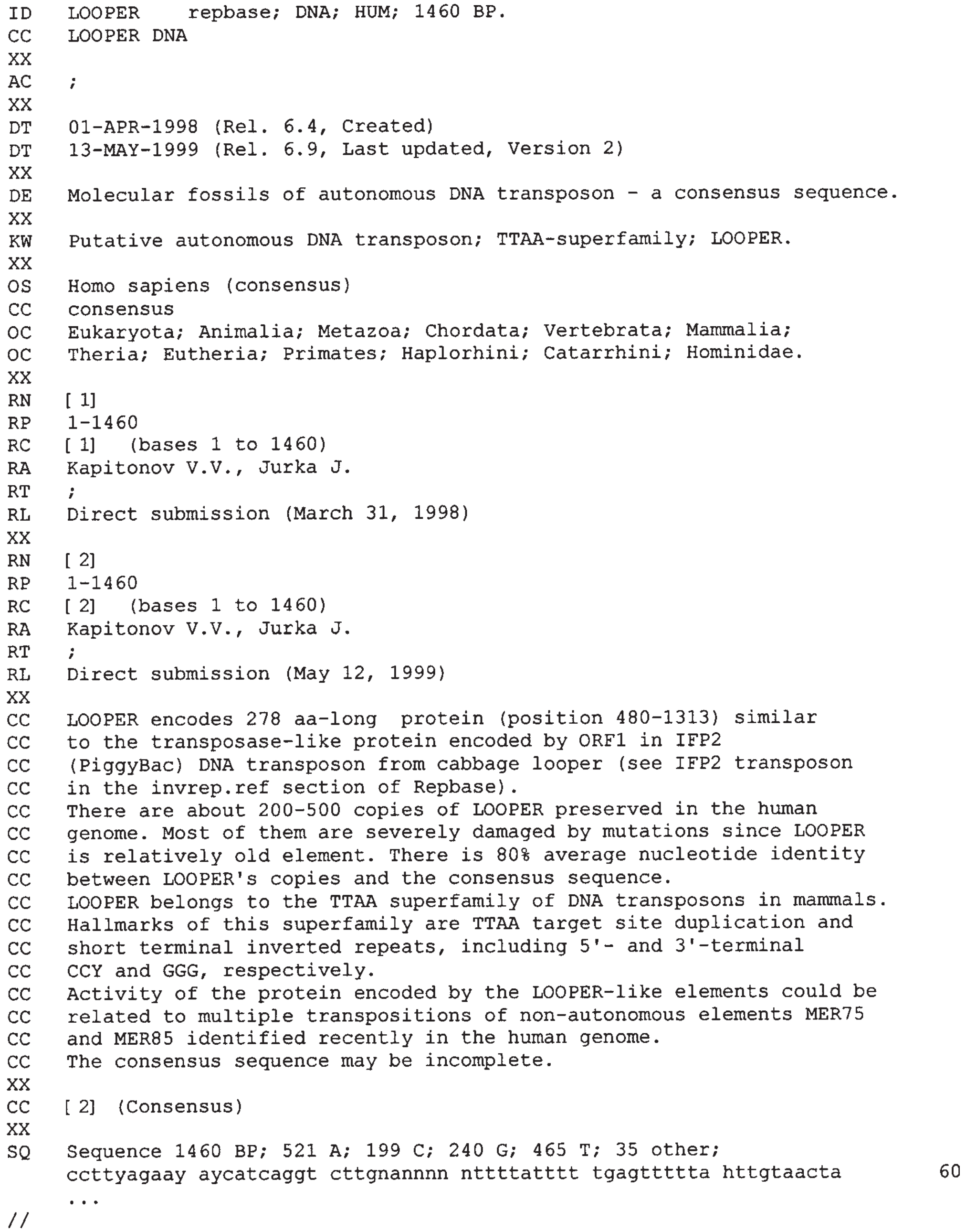
It appears that all eukaryotic genomes integrated a patchwork of TEs inserted at different times from the beginning of their evolutionary history. As indicated earlier, the human genome is among the best repositories of repetitive DNA going back over 200 million years. Unlike humans, all repetitive elements in plants and insects appear to be relatively young. This may indicate a rapid turnover of TEs in plant and insect genomes.

Reference Collections of Repeats

A practical approach to identifying and masking repetitive DNA began with creating comprehensive reference collections of repeats that could be compared against newly sequenced DNA. Prior to whole-genome sequencing projects, only human sequences were available in sufficient quantities to reveal a significant variety of human repeat families. These studies laid the foundation for the first collection of 53 representative human repeats. It was followed by collections of other mammalian repeats and placed in a database named *Repbase*. Since 1997, Repbase was succeeded by *Repbase Update* and over time it included repeats from other eukaryotic species as they became available. Originally, Repbase Update (RU) played the role of a database and an electronic journal releasing newly discovered repetitive families that were not published elsewhere. As of September 2001, all previously unreported families are first published in a peer-reviewed electronic journal entitled Repbase Reports (*see* Website: http://www.girinst.org), and subsequently released to RU. This arrangement is designed to facilitate proper referencing and documentation of the original data deposited in RU.

Current Content of Repbase Update

The current release of RU contains around 2400 unique entries from all sequenced eukaryotic species. The primary release of RU is in the EMBL format. A sample entry is shown in Fig. 2. Simultaneously, RU is also released in fasta format (without annotations) as well as in preprocessed, software-specific RepeatMasker format. The major files in the current release include repeats from humans (humrep.ref), rodents (rodrep.ref), other mammals (mamrep.ref), Zebrafish (zebrep.ref), other vertebrates (vrtrep.ref), *Caenorhabditis elegans* (celrep.ref), *Drosophila melanogaster* (drorep.ref), other animals (invrep.ref), *Arabidopsis thaliana* (athrep.ref), other plants and fungi (plnrep.ref), and simple repeats (simple.ref). Some sections can be merged together, pending specific needs. For example, the reference collection of plant repeats used by CENSOR server (*see* Website: http://www.girinst.org) includes two RU files: athrep.ref and plnrep.ref.



```
ID   LOOPER    repbase; DNA; HUM; 1460 BP.
CC   LOOPER DNA
XX
AC   ;
XX
DT   01-APR-1998 (Rel. 6.4, Created)
DT   13-MAY-1999 (Rel. 6.9, Last updated, Version 2)
XX
DE   Molecular fossils of autonomous DNA transposon - a consensus sequence.
XX
KW   Putative autonomous DNA transposon; TTAA-superfamily; LOOPER.
XX
OS   Homo sapiens (consensus)
CC   consensus
OC   Eukaryota; Animalia; Metazoa; Chordata; Vertebrata; Mammalia;
OC   Theria; Eutheria; Primates; Haplorhini; Catarrhini; Hominidae.
XX
RN   [ 1]
RP   1-1460
RC   [ 1]  (bases 1 to 1460)
RA   Kapitonov V.V., Jurka J.
RT   ;
RL   Direct submission (March 31, 1998)
XX
RN   [ 2]
RP   1-1460
RC   [ 2]  (bases 1 to 1460)
RA   Kapitonov V.V., Jurka J.
RT   ;
RL   Direct submission (May 12, 1999)
XX
CC   LOOPER encodes 278 aa-long  protein (position 480-1313) similar
CC   to the transposase-like protein encoded by ORF1 in IFP2
CC   (PiggyBac) DNA transposon from cabbage looper (see IFP2 transposon
CC   in the invrep.ref section of Repbase).
CC   There are about 200-500 copies of LOOPER preserved in the human
CC   genome. Most of them are severely damaged by mutations since LOOPER
CC   is relatively old element. There is 80% average nucleotide identity
CC   between LOOPER's copies and the consensus sequence.
CC   LOOPER belongs to the TTAA superfamily of DNA transposons in mammals.
CC   Hallmarks of this superfamily are TTAA target site duplication and
CC   short terminal inverted repeats, including 5'- and 3'-terminal
CC   CCY and GGG, respectively.
CC   Activity of the protein encoded by the LOOPER-like elements could be
CC   related to multiple transpositions of non-autonomous elements MER75
CC   and MER85 identified recently in the human genome.
CC   The consensus sequence may be incomplete.
XX
CC   [ 2] (Consensus)
XX
SQ   Sequence 1460 BP; 521 A; 199 C; 240 G; 465 T; 35 other;
     ccttyagaay aycatcaggt cttgnannnn nttttatttt tgagttttta httgtaacta       60
     ...
//
```

Fig. 2. This is a sample entry from RU that describes a new class of human DNA transposons, related to *PiggyBac* transposon in cabbage looper. Its sequence has been reconstructed from the genomic fossil record (*see* subheading “Identification, Reconstruction and Classification of New TEs”). The entry includes: a uniqe sequence identification name (ID); definition (DE), date of creation and of the latest update (DT), keywords (KW), biological classification of the species in which it was found (OC), reference (RN, RA, RL), basic commentary (CC), source of the sequence data (DR), and base composition of the consensus sequence (SQ). This particular consensus sequence led to identification of a PiggyBac-like gene in the human genome.

Sequences deposited in RU continue to be updated over time and the record of modified entries is preserved in appendix files accompanying the active files. For example, changes in the active file of human/primate repeats *humrep.ref* are documented in *humapp.ref*; changes in *rodrep.ref* are documented in *rodapp.ref*. Appendix files are for documentation purposes only, and they should not be used for annotation. Most repetitive elements deposited in RU have been assigned to general biological categories discussed earlier. For practical reasons, some of these categories are expanded within the framework of the original classification. For instance, mammalian retroviruses include *true* retrovirus-like elements and a distinct category of retroelements called *MaLRs*, distantly related to retroviruses. Moreover, long terminal repeats (LTRs) are listed separately from internal protein-coding retrovirus sequences. LTRs often represent the only known fragments of retroviruses in RU because they are by far more abundant and more readily identifiable than the internal sequences. Finally, except in the case of LINEs and SINEs, separation between autonomous and nonautonomous elements is not always possible because many elements in RU are reconstructed from inactive genomic copies. Therefore, nonautonomous and autonomous LTR-retroelements and DNA transposons are listed together.

Nomenclature

Each repetitive element listed in RU carries a unique name. Originally, standard names such as *medium reiteration frequency repeats* or *MERs* were assigned to unclassified sequence fragments discovered at the time. As more sequence information became available the nomenclature has evolved. For example, MER37 was later classified as a DNA transposon and became *Tigger*, MER115 and MER118 led to identification of *Zaphod* (*see* Repbase Update). The evolving nomenclature has been preserved in the keyword *KW* sections of RU entries (*see* Fig. 2), and in the appendix files. Unfortunately, new names usually are not any more informative than the old ones, as there are no standards that would systematically relate particular names to biological classification. Entire genomes have been sequenced and annotated based on Repbase Update. The value of such annotation depends to a large extent on the ability of non-specialists to rapidly classify particular repeats based on their names. This requires comprehensive indexes linking individual names, or groups of names from RU to the corresponding classes of repeats. Because repeat annotation in the human genome is based exclusively on RU and the reference collection of repeats is among most complete in RU, the first such index was prepared for human TEs. This index is expanded here (*see* Table 1 on pages 157 and 158) to include the nomenclature of other mammalian repeats from RU. Table 1 includes three columns. Column 3 summarizes individual and group names of repetitive families/subfamilies as used in RU. Column 1 lists major biological categories discussed in the previous section. Column 2 reflects further subdivision of TEs based on biological attributes as well as on their occurrence in different mammalian species. It must be noted here that many repetitive elements are shared among different mammals (e.g., MIR, MIR3) or are closely related (e.g., L1 elements). For practical purposes, the shared elements are listed only once in Table 1. For example, MIR, MIR3, and L3 present in all mammals, are listed only under *human/shared* category. This category also contains human-specific sequences such as Alu, SVA, and SVA2.

The current release of RU contains 835 mammalian repeat families and subfamilies, many of which share variants of the same name (e.g., MT2A, MT2B, MT2C).

Table 1
Major Categories of Mammalian Repetitive Families from Repbase Update and Their Family/Subfamily Loci Names

Major categories	*Subcategories*	*Family/Subfamily loci names*
LINEs	Human/shared	IN25, L1*, L2A, KER2, L3, CR1_HS
	Rodent/shared	LINE3_RN (L1_RN), LX*, LLME
	Other mammalian	ARMER1, ART2*, BDDF*, BOV2, BTALU2, LINE1E_OC, LINE_C, THER2
SINEs	Human/shared	Alu*, FLA*, HAL1*, L2B, MIR, MIR3, SVA, SVA2
	Rodent/shared	B1*, B2*, B3*, BC1, FAM, ID_B1, MUSID* (ID1-6), PB1*, RDRE1_RN (ID_RN), RNALUIII, RSINE1, RSINE2*(B4*), SQR2_MM
	Other mammalian	BCS, BOVA2, BOVTA, BTALU1, BTCS, C_OC, CHR-1, CHR-2, D_CH, DRE1, MAR1, MON1, MVB2, NLA, PRE1_SS, SINEC*, THER1
Retroviruses and retrovirus-like elements (*internal sequences*)	MaLRs- human/shared	MLT1R, MLT1AR (MLT-int), MLT1CR (MLT1-int), MLT1FR (MLT1F-int), MSTAR (MST-int), THE1BR
	MaLRs- rodent/shared	MTAI (MT-int), ORR1AI (ORR1-int), ORR1BI (ORR1B-int).
	Other human/shared retroviruses	ERVL, HARLEQUIN, HERV*, HRES1, HUERS-P*, LOR1I, MER4I, MER4BI, MER21I, MER31I, MER41I, MER50I, MER51I, MER52AI, MER57I, MER57A_I, MER61I, MER65I, MER66I, MER70I, MER83AI, MER83BI, MER84I, MER89I, MER110I, PABL_AI, PABL_BI, PRIMA4_I, PRIMA41
	Other rodent/shared retroviruses	ETNERV, IAPA_MM, IAPEYI, IAPEZI, MERVL, MMETN, MMLV30, MULV, MYS1_PL (MYSPL), MYSERV
Long terminal repeats (LTRs) *a*	MaLR LTRs- human/shared	MLT1*, MST*, THE1*
	MaLR LTRs- rodent/shared	MTA, MTB, MTC, MTD, MTE, MT2*, ORR1A*, ORR1B*, ORR1C, ORR1D
	Other human/shared retrovirus LTRs	HARLEQUINLTR, LTR*, LOR1, MLT2*, MER4*, MER9, MER11*, MER21*, MER31*, MER34*, MER39*, MER41*, MER48, MER49, MER50*, MER51*, MER52*, MER54*, MER57*, MER61*, MER65*, MER66*, MER67*, MER68*, MER70*, MER72*, MER73, MER74*, MER76, MER77, MER83*, MER84, MER87, MER88, MER89, MER90, MER92*, MER93*, MER95, MER101*, MER110*, PABL_A, PABL_B, PRIMA4_LTR, PTR5

(continued on next page)

Table 1 *(continued)*

Major categories	*Subcategories*	*Family/Subfamily loci names*
	Other rodent/shared retrovirus LTRs	BGLII, LTRIAPEY, LTRIS, MERVL_LTR, MYS1_LTR, NICER_RN, PMR89, RAL, RLTR*, RMER2,RMER3, RMER4, RMER5, RMER6*, RMER10, RMER12, RMER13*, RMER15, RMER16, RMER17*, RMER19,RMER20
	Other mammalian retrovirus LTRs	ALTR2, BTLTR1, ECE1LTR, FCLTR1, MTV9LTR1_SM
DNA Transposons	Human/shared	BLACKJACK, CHARLIE*, CHESHIRE*, GOLEM*, HSMAR*, HSTC2, LOOPER, MADE1, MARNA, MER1*, MER2*, MER3, MER5*, MER6*, MER8, MER20*, MER28, MER30*, MER33, MER44*, MER45*, MER46, MER53, MER63*, MER69*, MER75, MER80, MER81, MER82, MER85, MER91*, MER94, MER96*, MER97*, MER99, MER103, MER104*, MER105, MER106*, MER107, MER113, MER115, MER116, MER117, MER119, ORSL, PMER1, RICKSHA*, TIGGER*, ZAPHOD, ZOMBI*
	Rodent/shared	URR1
Minisatellites Satellites	Human	IVR, R66
	Human/shared	ALR*, BSR, CER, (GGAAT)n, GSAT*, HSATI*, LSAU, MER22, MER122, MSR1, REP522, SAR, SATR*, SN5, TAR1
	Rodent/shared	CENSAT, GSAT_MM, ISAT_RN, R91ES8_RN, SATI_RN, SATMIN, ZP3AR_MM
	Other mammalian	BMSAT1, BTSAT*, FASAT(?), OOREP1(L1?), OSSAT*, SSRS2, RTREP1, SATIA_MM
Composite/simple	Human	MER120
	Rodent	SQR1_MM
	Other mammalian	MRSAT1, SSRS1
Unclassified/ incomplete	Human	HIR, MER35, MER109, MER112, MER121
	Rodent	ALSAT_RN (L1?), C573, CYRA11_MM, DRB_RN, LPKR_RN, MREP_MC, PMR89 (HERVL?), RMER1*, SQR4_MM, YREP
	Other mammalian	LMER1

a This list does not include the internal sequence names listed above.

* Indicates multiple names starting with the same theme name. For example, RMER1* represents three subfamilies named RMER1, RMER1A, and RMER1B. The theme name *RMER1* is shared among the three.

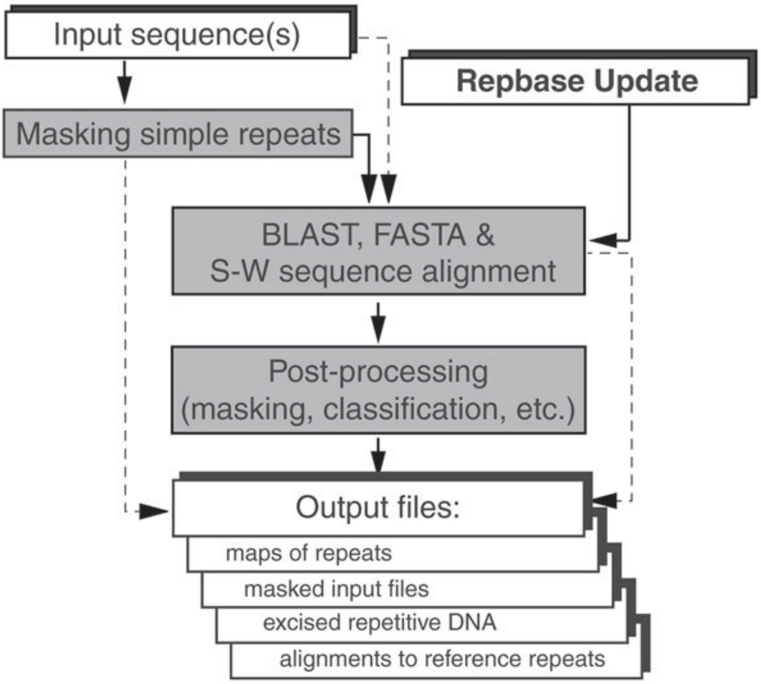


Fig. 3. A scheme for automated identification and annotation of repetitive DNA. Continuous arrows indicate critical steps, whereas broken arrows show major variants of the process. Typical output files include: maps of repeats (*see* Fig. 4), masked query file(s), a list of masked sequences and alignments against the reference sequences as described in the text.

Such variants are grouped together and the variations are indicated by asterisks (MT2*, *see* column 3). For example, RLTR* stands for 27 different long terminal repeats (LTRs). For historical reasons LTRs also carry other unassuming names such as BGLII, PMR89, and RMER. The corresponding internal retroviral sequences, if available, are listed in a separate section above the LTR section in Table 1. The application of Table 1 to the interpretation of repeat annotation is discussed in the next section.

Analysis of Repetitive DNA

Identification and Annotation of Known Repeats

The basic routine underlying identification and annotation of repetitive DNA remains essentially unchanged since it was first implemented in the Pythia server, and re-implemented in XBLAST and CENSOR. Since 1996, major progress has been achieved in terms of speed and sensitivity of repeat detection based on dedicated hardware used in CENSOR server (*see* Website: http://www.girinst.org/Censor_Server.html), and an efficient implementation of Smith-Waterman algorithm used in RepeatMasker (*see* Website: http://repeatmasker.genome.washington.edu).

Detection and annotation of repetitive DNA is based on comparing a query sequence against representative collections of repeats as schematically shown in Fig. 3.

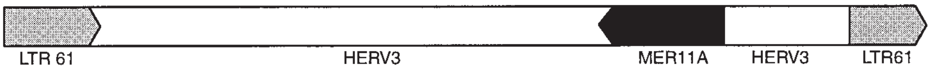
To avoid nonspecific matches, it is first advisable to filter out simple repeats from the query or reference sequences prior to the analysis by replacing sequence letters with neutral characters such as "N" or "X." There are several ways to identify simple repeats based on their similarity to a reference set, or on their non-random base composition. Another program, particularly useful for analyzing cryptically simple repeats, has been implemented as a part of repeat analysis on the CENSOR server (*see* Website: http://www.girinst.org/Censor_Server.html). After simple repeats are masked, *complex* repeats can be detected by sequence comparisons using the FASTA (1988), BLAST (1990), or Smith-Waterman (1981) algorithms. FASTA and BLAST are significantly faster but less sensitive than Smith-Waterman-based programs. The latter are essential to detect very old repetitive elements such as the extinct human MIR3 and LINE3 elements that are related to CR1 elements from birds. Conversely, relatively young repeats such as most Alu subfamilies can be detected using a less sensitive approach. Therefore, dividing repeats into detectable categories and the selective application of different algorithms may facilitate the detection process. Apart from these knowledge-based improvements, there are algorithm-based attempts to accelerate repeat detection without sacrificing the sensitivity of the process. It must be noted, the major determinant of speed, sensitivity, and accuracy is the quality of reference collections as discussed in the next section.

There are several types of output files generated by repeat annotation programs (examples are listed in Fig. 3). They include *maps of repeats* summarizing location and basic characteristics of individual elements, query file (s) with masked repeats, sequences and coordinates of the identified repeats and alignment to reference sequences for detailed inspection. A sample of repeat maps generated by CENSOR (*see* Website: http://www.girinst.org/Censor_Server.html) is shown in Fig. 4. The top part of the figure (Fig. 4A), illustrates a human endogenous retrovirus HERV3 flanked at both ends by long terminal repeats (LTR61). It also contains a MER11A element inserted in opposite orientation. All MER11 elements are classified as LTRs different from LTR61 (*see* Table 1). Therefore, MER11A most likely represents a remnant from a different retrovirus inserted at that spot. The internal portion of the retrovirus (HERV3) and the 5' LTR are chopped by the program into smaller fragments separated by *blank regions* without any identified repeats. Because the blank regions are relatively short (<50 bp), the fragments can be combined into a single LTR (see the bottom part of the figure). If the bank regions are much longer in relative terms, additional testing may be required. For example the first two 5' fragments of the internal sequence HERV3 are separated by 457 bp. A corresponding gap of similar size can be seen in the reference sequence from RU (*see* columns 4–6). This suggests that the nucleotide sequence at positions 54,059–54,515 represents another homologous fragment of the same retrovirus not detected by the algorithm. In such cases a separate test may be needed to verify whether or not such a spot does not represent any unknown element (s) inserted in the retrovirus. In this case, additional alignment has shown that this region is most similar to HERV3 and, therefore, it can be incorporated in the internal portion of the retrovirus pictured below the map.

The bottom example (Fig. 4B), shows a more complicated pattern of insertion into another endogenous retrovirus (HERVK22I), flanked by LTR22. This retrovirus contains two independent L1 elements (L1PA2 and L1PA3) inserted in its internal sequence. L1PA2 appears to be complete or nearly complete whereas L1PA3 is repre-

A

GB Acc.	Beg.	End	RU Locus	Beg.	End	O	S
AC007379	53088	53232	LTR61	74	233	d	0.74
AC007379	53260	53333	LTR61	234	311	d	0.76
AC007379	53368	53661	LTR61	299	595	d	0.77
AC007379	53671	54058	HERV3	10	393	d	0.74
AC007379	54515	54846	HERV3	794	1126	d	0.69
AC007379	55021	55802	HERV3	1302	2081	d	0.73
AC007379	55925	56582	HERV3	2161	2826	d	0.71
AC007379	56638	57091	HERV3	2863	3305	d	0.72
AC007379	57108	57996	HERV3	3334	4221	d	0.77
AC007379	58043	58767	HERV3	4268	5005	d	0.73
AC007379	58861	59905	MER11A	1126	1	c	0.86
AC007379	60087	61081	HERV3	5278	6313	d	0.72
AC007379	61173	61305	HERV3	8287	8418	d	0.74
AC007379	61316	61829	LTR61	72	595	d	0.74



B

GB Acc.	Beg.	End	RU Locus	Beg.	End	O	S
AB045363	38248	38745	LTR22	1	580	c	0.70
AB045363	38748	40067	HERVK22I	5514	6835	c	0.89
AB045363	40075	40494	L1PA3	483	902	c	0.98
AB045363	40495	42621	HERVK22I	5527	2327	c	0.86
AB045363	43041	43432	HERVK22I	339	741	c	0.85
AB045363	43434	48611	L1	1	5303	d	0.95
AB045363	48612	49463	L1PA2	51	902	d	0.97
AB045363	49492	49842	HERVK22I	3	350	c	0.88
AB045363	49845	50338	LTR22	1	580	c	0.69

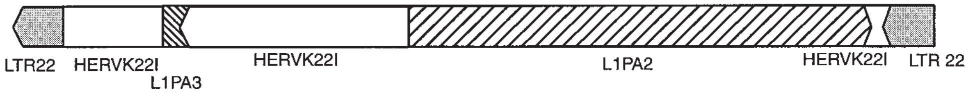


Fig. 4. Sample maps of repetitive elements generated by CENSOR (*see* Website: http://www.girinst.org), and their graphic interpretation. Column 1 lists the GenBank accession numbers of the query sequences, followed by coordinates of the repeats. Column 4 lists repeat names and coordinates relative to RU sequences. Column 7 shows the orientation of the repeats (direct or complementary), and the last column shows similarities to sequences from RU. Graphic interpretations are given below the maps. **(A)** HERV3 flanked by LTR61. **(B)** HERVK221 flanked by LTR22.

sented only by a short 419 bp 3' fragment and its orientation is opposite to L1PA3. Due to this opposite orientation, and typical 5' truncation, it is likely that L1PA3 represents a separate integration event of an incomplete L1 element.

Identification, Reconstruction and Classification of New TEs

Interspersed repetitive DNA seldom contains complete copies of TEs. It is believed that from the start many retro (trans)posons generate 5' truncated copies of themselves (e.g., L1PA3 in Fig. 4B). Even if the inserted copies are originally complete, over time they can undergo genetic rearrangements or partial to complete deletion. Furthermore, copies of TEs undergo base-substitution mutations, accelerated by methylation of CpG dinucleotides, due to conversion of the 5-methylcytosine to thymine by spontaneous deamination. As a result methylated CpG doublets mutate to CpA or TpG at a rate about a factor of magnitude higher than the average mutation rate. Thus, fragmented and mutated copies usually represent the only source of information about complete and active TEs. For this reason most TEs deposited in RU represent consensus sequences reconstructed from scattered partial sequence information. Consensus sequences are not only more complete, but also more similar to individual repeats than individual repeats to each other. The relationship between similarity to the consensus (y) and average pairwise similarity between individual repeats (x) is given by the Equation 1:

$$y = \frac{1 + \sqrt{12x - 3}}{4} \qquad \text{[Eq. 1]}$$

For example, individual repeats that are on average 50% similar to each other will be 68% similar to their quality consensus sequence (i.e., for x = 0.5, y = 0.68). This significantly facilitates the identification of highly diverged repeats.

The reconstruction process is usually, but not necessarily, associated with discovering new repetitive elements. There are many different ways in which previously unknown repetitive elements can be identified. For example, the coordinates of potential new repeat sequences can be determined from *blank spots* in the maps similar to those described above. A routine approach, presented in Fig. 5, takes advantage of the same computer software as used in annotating known repeats. It starts with masking the existing repeats from a Genbank file or other large data set. The masked sequence data is then compared against itself, preferably in opposite orientation to minimize obvious matches from known multi-copy genes. The resulting output file will contain homologous sequence fragments including repeats. These repeats need to be tested to exclude matches between duplicated genes or other obvious similarities not related to repetitive DNA. The next critical step is the generation of a multiple alignment as a basis for developing a consensus as summarized in Fig. 5. Multiple alignment of partial sequences can be carried out using CLUSTALX program (*see* Chapter 31), but it usually requires refinement using sequence editors. A partial consensus sequence may be used to realign sequence fragments in order to verify and improve the original alignment. This iterative approach is particularly useful in the case of highly diverged sequences.

Prior to consensus building it is important to subdivide repetitive elements into meaningful subfamilies if the number of repeat sequences is sufficient. In most cases it can be done using a standard phylogenetic analysis package (e.g., PHYLIP; *see* Website: http://evolution.genetics.washington.edu/phylip.html). In the past, constructing phylogenetic trees proved to be unsuccessful in determining Alu classification, and alternative approaches were necessary. Retrospectively, most difficulties encountered using tree-based classification for repeats were due to interference between rapidly mutating CpG doublets and other sequence positions containing critical information. Therefore, in the case of at least CpG-rich repeats, it may be important to exclude CpG doublets and their common derivatives (TpG, CpA) when constructing phylogenetic trees.

Initial reconstruction almost always leads to partial consensus sequences. To determine exact boundaries, it is important to obtain maps of repeats that would include the newly built consensus sequence. The best way to determine exact ends of the new

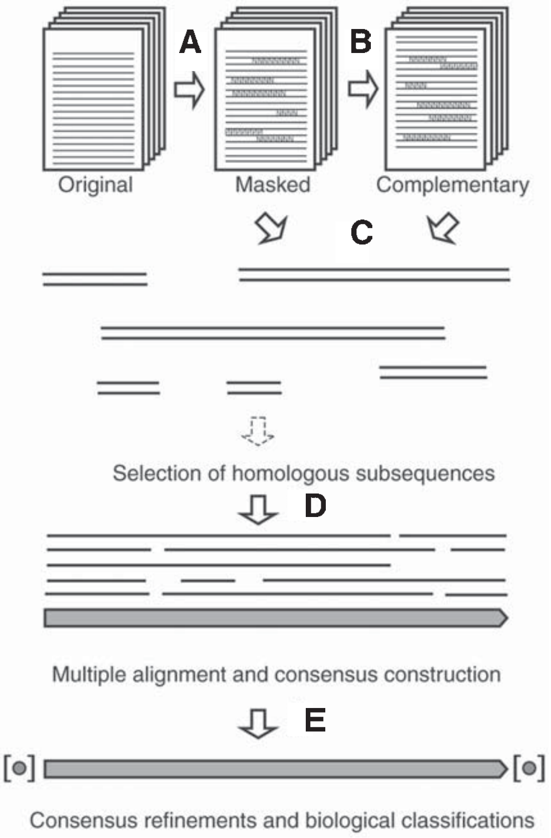

A
B
Original
Masked
Complementary
C
Selection of homologous subsequences
D
Multiple alignment and consensus construction
E
Consensus refinements and biological classifications

Fig. 5. Identification and reconstruction of TEs. The steps involved are: **(A)** masking known repeats in the query database; **(B)** generation of a complementary database; **(C)** comparison of direct and complementary databases in search of homologous regions, followed by selection of potential repeats; **(D)** multiple alignment and consensus building; **(E)** evaluation of consensus and determination of target sites as detailed in the text.

repeat is to study its insertions into other repetitive elements. This is also the best way to determine target site duplications that are often essential for proper classification of the repeat.

Systematic analysis of sequence data during the last decade has revealed a large variety of transposable elements in eukaryotic genomes. This has led to the creation of specialized databases and tools essential for eukaryotic genome analysis during the sequencing era. Some approaches outlined in this chapter are relatively straightforward. However, much of the analysis still relies heavily on human judgement and creativity that have not yet been encoded in computer software. Transposable elements are intrinsically involved in biological processes that still remain to be understood. Therefore, repetitive DNA is no longer viewed as a troublesome *junkyard* but rather as a gold mine of information underlying biology of eukaryotic genomes.

Acknowledgment

I thank Jolanta Walichiewicz, Violaine Genot, and Michael Jurka for help with editing the manuscript.

Suggested Readings

Introduction

Andersson, G., Svensson, A. C., Setterblad, N., and Rask, L. (1998) Retroelements in the human MHC class II region, Trends Genet. 14, 109–114.

Hartl, D. L. (2000) Molecular melodies in high and low C, Nature Genet. Rev. 1, 145–149.

Jabbari, K. and Bernardi, G. (1998) CpG doublets, CpG islands and Alu repeats in long human DNA sequences from different isochore families, Gene 224, 123–128.

Jurka, J. (1990) Novel families of interspersed repetitive DNA sequences from the human genome, Nucleic Acids Res. 18, 137–141.

Jurka, J. (1998) Repeats in genomic DNA: mining and meaning, Curr. Opin. Struct. Biol. 8, 333–337.

Lander, E. S., Linton, L. M., Birren, B., et al. (2001) Initial sequencing and analysis of the human genome, Nature 409, 860–921.

Sherry, S. T., Harpending, H. C., Batzer, M. A., and Stoneking, M. (1997) Alu evolution in human populations: using the coalescent to estimate effective population size, Genetics 147, 1977–1982.

Shimamura, M., Yasue, H., Ohshima, K., Abe, H., Kato, H., Kishiro, T., et al. (1997) Molecular evidence from retroposons that whales form a clade within even-toed ungulates, Nature 388, 666–670.

Surzycki, S. A. and Belknap, W. R. (1999) Characterization of repetitive DNA elements in Arabidopsis, J. Mol. Evol. 48, 684–691.

Simple Sequence Repeats (SSRs)

Hancock, J. M. (1995) The contribution of slippage-like processes to genome evolution, J. Mol. Evol. 41, 1038–1047.

Jeffreys, A. J., Barber, R., Bois, P., Buard, J., Dubrova, Y. E., Grant, G., et al. (1999) Human minisatellites, repeat DNA instability and meiotic recombination, Electrophoresis 20, 1665–1675.

Jurka, J. (1995a) Human Repetitive Elements, in: Molecular Biology and Biotechnology. A Comprehensive Desk Reference, (Meyers, R. A., ed.), VCH Publishers Inc., New York, NY, pp. 438–441.

Jurka, J. (1998) Repeats in genomic DNA: mining and meaning, Curr. Opin. Struct. Biol. 8, 333–337.

Kashi, Y., King, D., and Soller, M. (1997) Simple sequence repeats as a source of quantitative genetic variation, Trends Genet. 13, 74–78.

Kidwell, M. G. and Lisch, D. R. (2001) Perspective: transposable elements, parasitic DNA, and genome evolution, Evolution Int. J. Org. Evolution 55, 1–24.

Toth, G., Gaspari, Z., and Jurka, J. (2000) Microsatellites in different eukaryotic genomes: survey and analysis, Genome Res. 10, 967–981.

Weber, J. L. and May, P. E. (1989) Abundant class of human DNA polymorphisms which can be typed using the polymerase chain reaction, Am. J. Hum. Genet. 44, 388–396.

Transposable Elements (TEs)

Jurka, J. (1995a) Human Repetitive Elements, in: Molecular Biology and Biotechnology. A Comprehensive Desk Reference, (Meyers, R. A., ed.), VCH Publishers Inc., New York, NY, pp. 438–441.

Jurka, J. (1995b) Origin and evolution of Alu repetitive elements, in: The Impact of Short Interspersed Elements (SINEs) on the Host Genome, (Maraia, R. J., ed.), R. G. Landes Company, Austin, TX, pp. 25–41.

Jurka, J. (1997) Sequence patterns indicate an enzymatic involvement in integration of mammalian retroposons, Proc. Natl. Acad. Sci. USA 94, 1872–1877.

Jurka, J. and Kapitonov, V. V. (1999) Sectorial mutagenesis by transposable elements, Genetica 107, 239–248.

Reference Collections of Repeats

Jurka, J. (1990) Novel families of interspersed repetitive DNA sequences from the human genome, Nucleic Acids Res. 18, 137–141.

Jurka, J., Walichiewicz, J., and Milosavljevic, A. (1992) Prototypic sequences for human repetitive DNA, J. Mol. Evol. 35, 286–291.

Jurka, J., Kaplan, D. J., Duncan, C. H., Walichiewicz, J., Milosavljevic, A., Murali, G., and Solus, J. F. (1993) Identification and characterization of new human medium reiteration frequency repeats, Nucleic Acids Res. 21, 1273–1279.

Jurka, J. (2000) Repbase Update: a database and an electronic journal of repetitive elements, Trends Genet. 16, 418–420.

Kaplan, D. J. and Duncan, C. H. (1990) Novel short interspersed repeat in human DNA, Nucleic Acids Res. 18, 192.

Kaplan, D. J., Jurka, J., Solus, S. F., and Duncan, C. H. (1991) Medium reiteration frequency repetitive sequences in the human genome, Nucleic Acids Res. 19, 4731–4738.

Smit, A. F. (1993) Identification of a new, abundant superfamily of mammalian LTR-transposons, Nucleic Acids Res. 21, 1863–1872.

Smit, A. F., Toth, G., Riggs, A. D., and Jurka, J. (1995) Ancestral, mammalian-wide subfamilies of LINE-1 repetitive sequences, J. Mol. Biol. 246, 401–417.

Annotations Based on Repbase Update

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., et al. (2001) Initial sequencing and analysis of the human genome, Nature 409, 860–921.

Nomenclature

Iris, F., Bougueleret, L., Prieur, S., Caterina, D., Primas, G., Perrot, V., et al. (1993) Dense Alu clustering and a potential new member of the NFkappaB family within a 90 kilobase HLA class III segment, Nature Genet. 3, 137–145.

Jurka, J. (1990) Novel families of interspersed repetitive DNA sequences from the human genome, Nucleic Acids Res. 18, 137–141.

Jurka, J., Kaplan, D. J., Duncan, C. H., Walichiewicz, J., Milosavljevic, A., Murali, G., and Solus, J. F. (1993) Identification and characterization of new human medium reiteration frequency repeats, Nucleic Acids Res. 21, 1273–1279.

Jurka, J., Kapitonov, V. V., and Smit, A. F. A. (2003) Repetitive DNA, detection of, in: Encyclopedia of the Human Genome, (Cooper, D. N., ed.) Nature Publishing Group, New York, NY.

Kaplan, D. J. and Duncan, C. H. (1990) Novel short interspersed repeat in human DNA, Nucleic Acids Res. 18, 192.

Kaplan, D. J., Jurka, J., Solus, S. F., and Duncan, C. H. (1991) Medium reiteration frequency repetitive sequences in the human genome, Nucleic Acids Res. 19, 4731–4738.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., et al. (2001) Initial sequencing and analysis of the human genome, Nature 409, 860–921.

Smit, A. F. A. and Riggs, A. D. (1996) *Tiggers* and other DNA transposon fossils in the human genome, Proc. Natl. Acad. Sci. USA 93, 1443–1448.

Analysis of Repetitive DNA

Identification and Annotation of Known Repeats

Bedell, J. A., Korf, I., and Gish, W. (2000) MaskerAid: a performance enhancement to RepeatMasker, Bioinformatics 16, 1040–1041.

Claverie, J. M. (1994) Large scale sequence analysis, in: Automated DNA Sequencing and Analysis, (Adams, M. D., Fields, C., and Venter, J. C., eds.), Academic Press, San Diego, CA, pp. 267–279.

Karlin, S. and Altschul, S. F. (1990) Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes, Proc. Natl. Acad. Sci. USA 87, 2264–2268.

Jurka, J. and Milosavljevic, A. (1991) Reconstruction and analysis of human Alu genes, J. Mol. Evol. 32, 105–121.

Jurka, J., Walichiewicz, J., and Milosavljevic, A. (1992) Prototypic sequences for human repetitive DNA, J. Mol. Evol. 35, 286–291.

Jurka, J. (1994) Approaches to identification and analysis of interspersed repetitive DNA sequences, in: Automated DNA Sequencing and Analysis (Adams, M. D., Fields, C., and Venter, J. C., eds.), Academic Press, San Diego, CA, pp. 294–298.

Jurka, J. and Pethiyagoda, C. (1995) Simple repetitive DNA sequences from primates: compilation and analysis, J. Mol.Evol. 40, 120–126.

Jurka, J., Klonowski, P., Dagman, V., and Pelton, P. (1996) CENSOR-a program for identification and elimination of repetitive elements from DNA sequences, Comput. Chem. 20, 119–121.

Milosavljevic, A. and Jurka, J. (1993) Discovering simple DNA sequences by the algorithmic significance method, Comput. Applic. Biosci. 9, 407–411.

Milosavljevic, A. (1998) Repeat analysis, in: ICRF Handbook of Genome Analysis, vol. 2, (Spurr, N. K., Young, B. D., and Bryant, S. P., eds.), Blackwell Science Inc., Malden, MA, pp. 617–628.

Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison, Proc. Natl.Acad. Sci. USA 85, 2444–2448.

Silva, R. and Burch, J. B. (1987) Evidence that chicken CR1 elements represent a novel family of retroposons, Mol. Cell. Biol. 9, 3563–3566.

Smith, T. F. and Waterman, M. S. (1981) Identification of common molecular subsequences, J. Mol. Biol. 147, 195–197.

Wootton, J. C. and Federhen, S. (1996) Analysis of compositionally biased regions in sequence databases, Meth. Enzymol. 266, 554–571.

Identification, Reconstruction and Classification of New TEs

Bao, Z. and Eddy, S. R. (2002) Automated de novo identification of repeat sequence families in sequenced genomes, Genome Res. 12, 1269–1276.

Faulkner, D. V. and Jurka, J. (1988) Multiple Aligned Sequence Editor (MASE), Trends Biochem. Sci 13, 321–322.

Galtier, N., Gouy, M., and Gautier, C. (1996) SeaView and Phylo_win, two graphic tools for sequence alignment and molecular phylogeny, Comp. Appl. Biosci. 12, 543–548.

Ivics, Z., Hackett, P. B., Plasterk, R. H., and Izsvak, Z. (1997) Molecular reconstruction of Sleeping Beauty, a Tc1-like transposon from fish, and its transposition in human cells, Cell 91, 501–510.

Jurka, J. and Milosavljevic, A. (1991) Reconstruction and analysis of human *Alu* genes, J. Mol. Evol. 32, 105–121.

Jurka, J. (1994) Approaches to identification and analysis of interspersed repetitive DNA sequences, in: Automated DNA Sequencing and Analysis (Adams, M. D., Fields, C. and Venter, J. C., eds.), Academic Press, San Diego, CA, pp. 294–298.

Jurka, J. (1998) Repeats in genomic DNA: mining and meaning, Curr. Opin. Struct. Biol. 8, 333–337.

Jurka, J. and Smith, T. (1988) A fundamental division in the *Alu* family of repeated sequences, Proc. Natl. Acad. Sci. USA 85, 4775–4778.

Kapitonov, V. V. and Jurka, J. (2001) Rolling-circle transposons in eukaryotes, Proc. Natl. Acad. Sci. USA 98, 8714–8719.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucleic Acids Res. 22, 4673–4680.

9 Molecular Genetics of Disease and the Human Genome Project

Paromita Deb-Rinker and Stephen W. Scherer

Introduction

The haploid (n) human genome contains approx 3 billion nucleotides (or bases) of DNA strung amongst 23 chromosomes (*see* Fig. 1). The diploid (2n) complement, which consists of a haploid genome inherited from each parent, therefore comprises 46 chromosomes of 6 billion nucleotides of DNA, all contained within the cell nucleus. The same complement of DNA is found in every cell (except red blood cells) in the body. Mitochondrial DNA, which is a circular molecule of genetic material, 16,000 nucleotides long, is also part of the human genome. It is located outside the nucleus in the cytoplasm of the cell and encodes a small but important subset of human genes. Mitochondrial DNA is only transmitted from mothers to their offspring. On average, the human genome is 99.9% identical between any two individuals, with nucleotide differences existing only about 1 in every 1000 bases. Less than 5% of the genome contains genes or protein-coding regions. The remaining 95% (noncoding part) contains repetitive elements (*see* Chapter 8) and other sequences whose functions are not completely understood and is often referred to as junk DNA. These regions may play a role in maintaining the structural integrity of chromosomes. There at least 30,000–40,000 genes in the human genome, ranging in size less than 1 to 200 kilobases (kb), with the average size of a gene being 50 kb. General information on the human genome is provided in Table 1.

Genes are comprised of exons—regions that code for mature mRNA—and introns—intervening segments of DNA that are transcribed, but then cut out from the transcript during mRNA processing (*see* Fig. 2). There is no uniformity to the number or size of introns; this is the main reason why there is a vast range of gene sizes. Genes contain promoter sequences at their start (5' end). Typical promoters contain several DNA sequence motifs that bind regulatory proteins and control the level of transcription and the start position of the mRNA. Expression of tissue-specific genes are unique to individual or sets of tissues (muscle, brain, liver, etc.) in our bodies. There are also housekeeping genes that are expressed in all cell types because their products provide basic functions (*see* Table 2).

Simple and Complex Patterns of Inheritance

Every individual has two copies of each gene, one on each of the chromosomes. Owing to DNA sequence variation, there can be two or more alternative forms of a gene

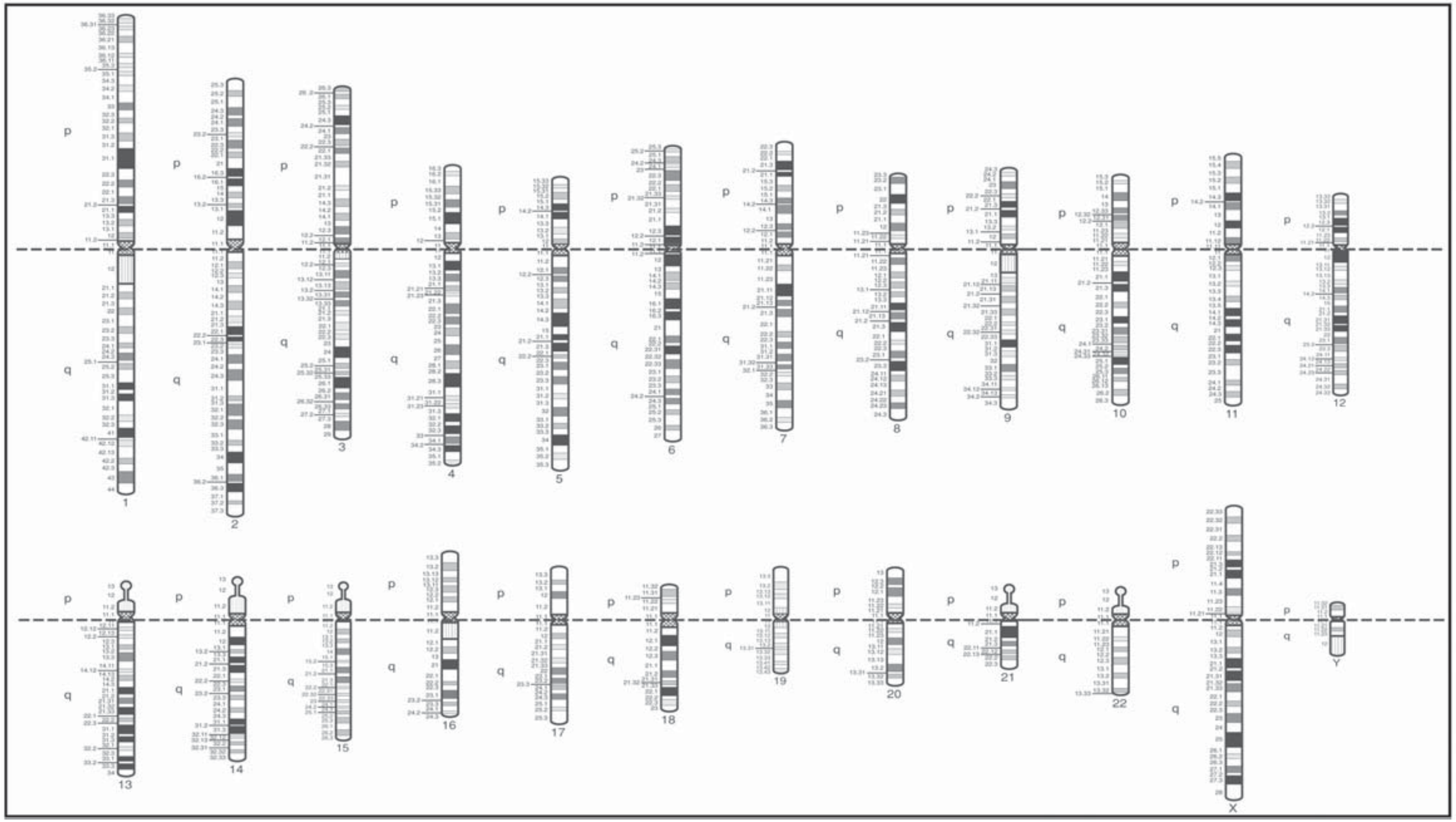


Fig. 1. The 23 chromosomes in the human genome. There are 22 autosomes (chromosomes 1 to 22) and two sex chromosomes (X and Y). Females inherit 22 autosomes and one X chromosome from each parent (46;XX). Males inherit 22 autosomes from each parent, an X chromosome from the mother and a Y chromosome from the father (46;XY). Metacentric chromosomes have a centromere (where spindle fiber binds during cell division) in the middle. Acrocentric chromosomes (chromosomes 13, 14, 15, 21, and 22) have their centromere near the end. The lighter bands of the chromosomes represent regions that have a higher GC-nucleotide content and more genes than the darker bands. Housekeeping genes tend to be located within the light bands and tissue-specific genes in the dark bands. The distribution of genes between chromosomes is also not uniform. For example, chromosome 21 is relatively gene-poor, while chromosome 19 is extremely gene-rich.

Table 1
Components of the Human Genome

- The genome is the total genetic material in a cell.
- The nuclear genome is comprised of 46 chromosomes, which come as 23 pairs; one of each pair comes from either parent.
- Mitochondrial DNA is also part of the genome. It is always inherited from the mother and is found in the cytoplasm.
- Chromosomes are made of deoxyribonucleic acid (DNA).
- DNA is made of four chemical units (nucleotides) called adenine (A), guanine (G), cytosine (C), and thymine (T).
- The genome is comprised of about 3 billion A, C, G, and Ts.
- Genes are the portions of DNA that encode functional RNA molecules or proteins.
- There are approx 30,000–40,000 genes in the genome.
- Proteins provide the structure for the cell and are involved in biochemical reactions (enzymes).

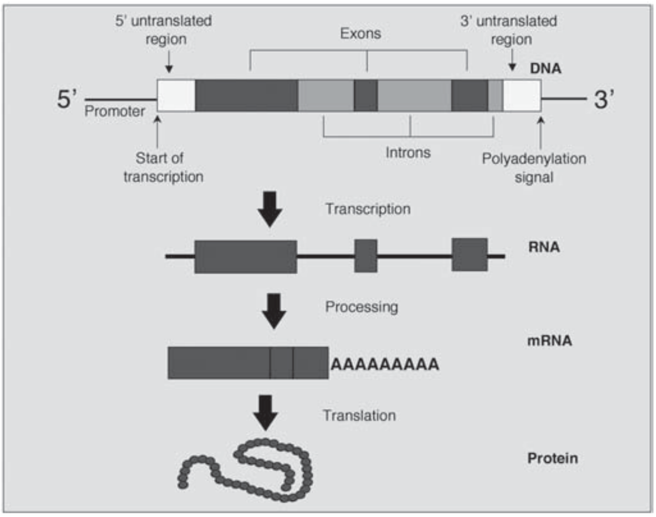


Fig. 2. Anatomy of a gene. Information flows from DNA to RNA (transcription) to protein (translation).

(called alleles), that result in different gene products. This variation contributes to the uniqueness of individuals. The term genotype is defined as the complete heritable genetic composition of an individual. Phenotype is the physical or biochemical manifestation of that genotype, and in some instances this can be associated with disease. However, in most cases, different alleles contribute to physical characteristics such as height, hair color, and other nondisease related cellular functions. The dominant and recessive forms of a trait in any given person are governed by which alleles (dominant or recessive) are inherited from the parents (*see* Fig. 3). Only individuals with two recessive alleles will show the recessive form of a trait (e.g., blue eyes). However, in the presence of a dominant allele (e.g., brown eyes), the trait associated with the recessive allele is not expressed phenotypically. Recessive alleles are not lost in a population; they can be passed on to subsequent generations where in the presence of another copy of the allele, the recessive trait reveals itself again.

See companion CD for color Fig. 3

Table 2
Gene Expression

Transcription	The synthesis of a single-strand RNA molecule from a DNA template in the cell nucleus. This process is controlled by the interactions between proteins and DNA sequences near each gene.
RNA processing	
Capping	Addition of a modified nucleotide chain to the 5' end of a growing mRNA chain. This is required for the normal processing, stability, and translation of mRNA.
Splicing	The process of removing introns and joining exons into a mature mRNA molecule.
Polyadenylation	Addition of 20–200 adenosine residues (poly A tail) to the 3' end of the RNA transcript.
Transport	The fully processed RNA is taken to the cytoplasm where translation takes place.
Translation	The synthesis of a protein from its mRNA template.
Housekeeping genes	Expressed in all cell types because their products provide basic functions in cells.
Tissue-specific genes	Expressed in only certain cell-types because their products have specific functions.

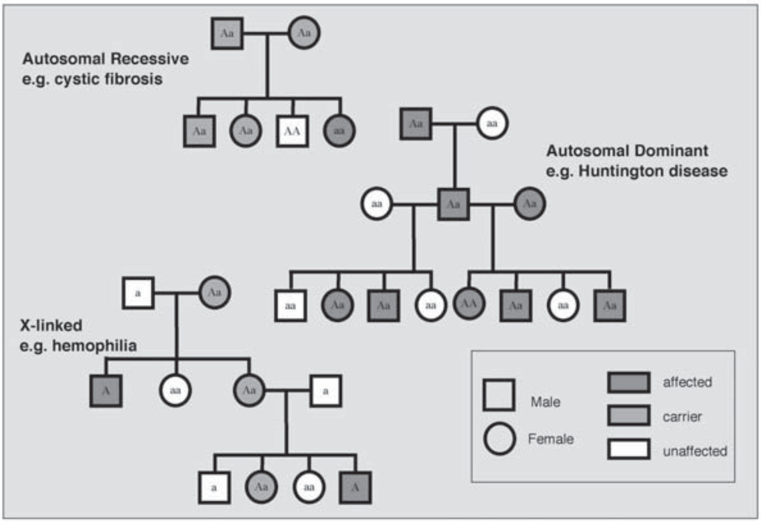


Fig. 3. Patterns of inheritance for autosomal recessive, autosomal dominant and X-linked disorders. Recessive alleles are indicated by 'a' and dominant alleles by 'A'. Males only have one X chromosome, therefore only one allele is shown for X-linked inheritance.

There are a number of diseases that are manifested owing to this simple dominant/recessive pattern of expression. Sickle cell anemia and cystic fibrosis are common examples of when the disease (termed autosomal recessive) develops owing to the presence of two copies of the recessive gene. A person with only one copy of the recessive allele does not develop the disease, but remains a carrier, because the normal copy of the gene predominates. Autosomal dominant disorders like Huntington's disease are produced when a single mutated dominant allele is present even if the other copy of the allele is normal. Diseases resulting from mutations in genes on the X chromosome are known as X-linked disorders. Since males only have one X chromosome, these diseases (e.g., hemophilia) act like dominant mutations in males. Females on the other hand, act as carriers and in the next generation their male offspring may or may not be affected (*see* Fig. 3).

Not all disorders and traits follow a simple pattern of inheritance as described earlier. One gene can influence more than one trait (pleiotropy) and several genes can affect only one trait (polygenic disorders). Although genes may determine whether or not a person will have heart disease or be predisposed to cancer, many traits can be triggered or influenced by the environment as well, as in the case of complex multifactorial diseases such as schizophrenia and alcoholism.

DNA Sequence Variations and Gene Mutations

The human genome is predominantly stable and does not vary significantly between individuals. As described earlier, the genome of two individuals is only 0.1% different. Some parts of the genome are more prone to variations than others, based on properties inherent to the DNA sequence. Most of these variations are found outside the coding regions of genes and are thus not harmful. These variations include mutations and polymorphisms.

A mutation is any permanent, heritable change in the DNA sequence that in some cases can result in disease. There are different kinds of genetic mutations. Gene mutations can be inherited from a parent or acquired during an individual's lifetime. The former, known as germline mutations, exists in the reproductive cells and can be passed on to later generations. This type of mutation is present in every cell that descended from the zygote to which the mutant gamete contributed. Somatic mutations are changes that occur in non-sex cells in the body (e.g., bone marrow or pancreatic cell), and are not passed on to the next generation. Whatever the effect, the ultimate fate of that somatic mutation is to disappear when the cell in which it occurred, or the individual, dies. On average, approx 200 somatic mutations accumulate within the genome during each of our lifetimes.

Genes can be altered in many ways (*see* Table 3). An incorrect base may be incorporated into the DNA sequence (point mutations) or a nucleotide or more may be left out (deletions) or added (insertions). In some diseases like cystic fibrosis, different mutations in the same gene can produce different effects in different individuals (*see* Fig. 4), but mutations in several different genes can also lead to the same clinical consequence, as in some forms of Alzheimer's disease. In some cases, the effects of mutational changes may not be critical to the proper functioning of proteins and these are called silent mutations. Other mutations can affect the structure and function of proteins. The outcome is related to the degree of change in the protein and the role a particular protein might play in the body. At times, longer stretches of DNA, millions of nucleotides long, can also be deleted or inserted. Occasionally, longer segments of DNA are doubled (duplications) or interchanged between chromosomes (translocations). The gain or loss of entire chromosomes or chromosomal segments can also occur. Common disorders associated with these kinds of chromosomal changes include Down's syndrome, trisomy 18 and trisomy 13. Patients with these diseases carry an extra chromosome 21, 18, or 13, respectively, in their cells.

Polymorphisms are differences in the genetic make-up of individuals that can be observed at the chromosomal, gene or even single base-pair level. They were originally considered to be genome variations that were neither harmful nor beneficial. However, there is evidence that polymorphisms can sometimes influence an individual's risk to disease. A single nucleotide polymorphism (SNP), which is a DNA point mutation, comes from variations in single nucleotides where any four of the bases may be substituted at a particular position. Most SNPs tend to have only two alleles instead of four. Considering that the human genome contains more than 3 million SNPs that make up 90% of polymorphisms and most of the genome variations, the trend towards genome-wide screening for insights into disease mechanisms is increasing.

Microsatellites (also called short tandem repeats, STRs) represent another class of genetic polymorphism. These are tandemly repeated sequences, where the repeating unit is 1–4 nucleotides long. The number of times the unit is repeated in a given STR can be highly variable, a characteristic that makes them useful as genetic markers. The majority of microsatellites occur in gene introns or other noncoding regions of the genome. Generally, the microsatellite itself does not cause disease; rather it is used as a marker to identify a specific chromosome or locus. When being used as a marker, the specific number of repeats in a given STR is not critical, but the difference in the number of repeats is of importance. There are other genetic markers that vary in length from 2–60 nucleotides; these are called variable number tandem repeats (VNTRs).

Table 3
Types of Mutations and Their Effects

Missense mutations	Single-nucleotide changes resulting in the substition of an amino acid in the protein.
Nonsense mutations	Single-nucleotide changes that create one of the three termination codons (TAA, TGA, or TAG) resulting in a shortened, dysfunctional protein.
Silent mutations	Have no detectable phenotypic effect.
Splice-site mutations	Altered sequences at the ends of introns (5' donor or 3' acceptor) during RNA processing, that affect gene splicing and function.
Insertions	Addition of extra DNA sequences of varying sizes.
Duplications	Doubling of DNA sequences.
Translocations	Interchange of segments of DNA between two different chromosomes.
Inversions	Occur when a region of DNA inverts its orientation with respect to the rest of the chromosome.
Trinucleotide repeats	Expansion of triplet repeat sequences.
Deletions	Loss of part of a DNA sequence (could be loss of a single nucleotide or millions of nucleotides).

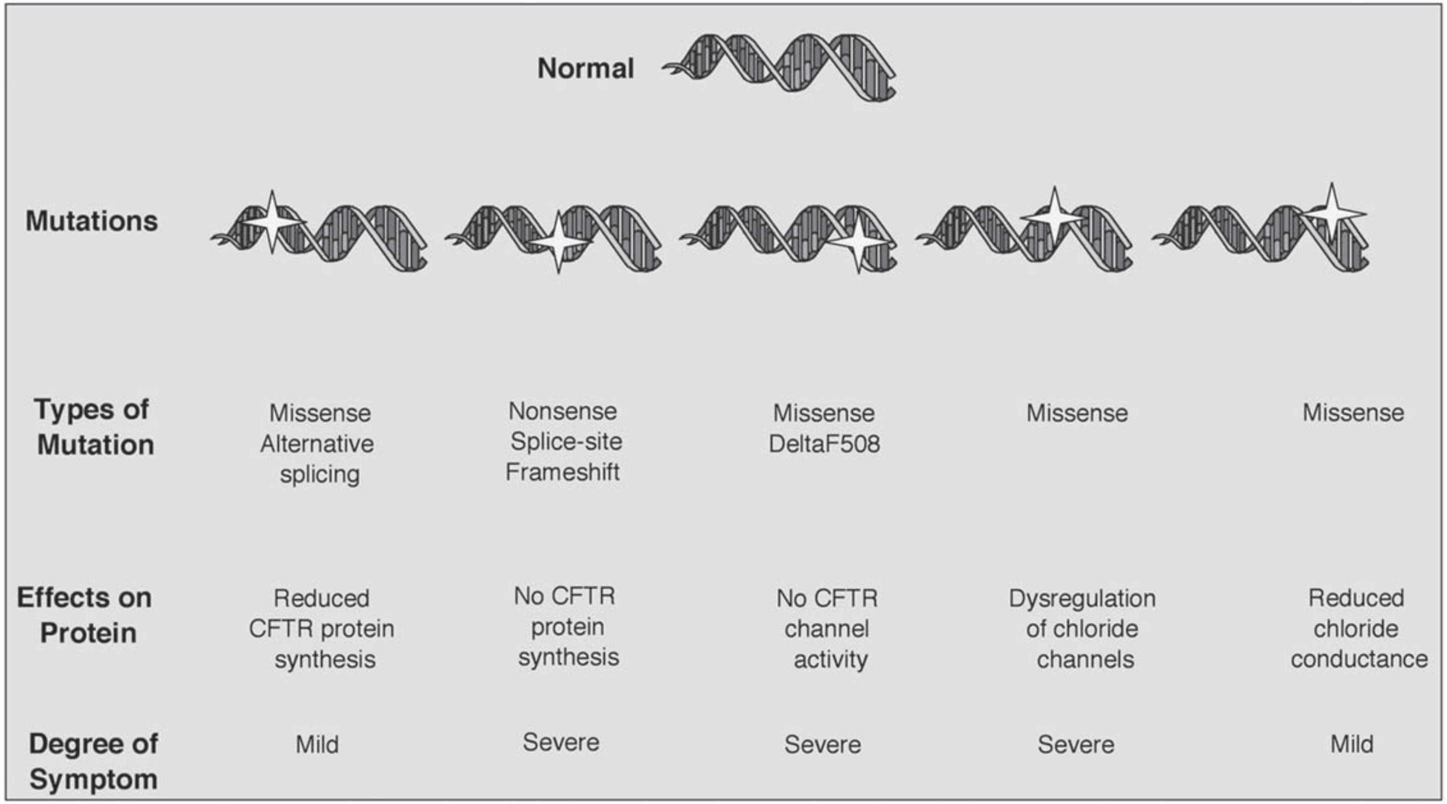
Deletion of:	Effects:
A gene	No protein product
An exon	Truncated protein
An intron	Usually no phenotypic change
Promoter	Gene nonfunctional
Splice-site	Protein nonfunctional
Many genes	Chromosomal abnormalities and usually a heterogeneous phenotype

The Human Genome Project and DNA Sequencing

The Human Genome Project (HGP) was formally initiated in the early 1990s, as an international effort to determine the complete DNA sequence of the human genome and all of the genes it encodes. The most immediate benefit of this information was to facilitate disease gene research.

DNA sequencing is the process of determining the specific order and identity of the three billion base pairs in the genome, with the ultimate goal of identifying all of the genes. Mapping is the process of identifying discrete DNA molecules of known position on a chromosome, which are then used for sequencing. Mapping is a crucial step for proper reconstruction of the genome. It usually precedes sequencing but is also necessary at the postsequencing stage.

Sequencing is now carried out through a process called fluorescence-based dideoxy sequencing. Fragments of DNA are first cloned in bacteria; they are then put into a polymerase catalyzed reaction with free nucleotides, some of which are tagged with fluorescent dyes. Nucleotides attach themselves to the DNA fragments in a particular order. Similarly, dyed nucleotides can attach themselves to the DNA fragments, but



Mutations					
Types of Mutation	Missense Alternative splicing	Nonsense Splice-site Frameshift	Missense DeltaF508	Missense	Missense
Effects on Protein	Reduced CFTR protein synthesis	No CFTR protein synthesis	No CFTR channel activity	Dysregulation of chloride channels	Reduced chloride conductance
Degree of Symptom	Mild	Severe	Severe	Severe	Mild

Fig. 4. Different mutations in the same gene (e.g., CFTR in cystic fibrosis) can produce different clinical outcomes. The clinical symptoms (severe/mild) correlate to the pancreatic function of the patients. Adapted from Zielenski and Tsui (1995).

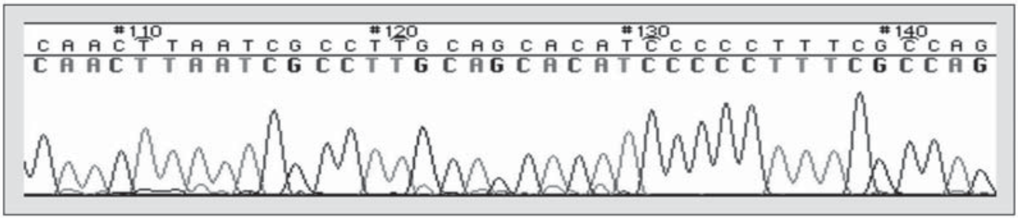


Fig. 5. DNA sequencing. Four different fluorescently tagged dyes (red for thymines, green for adenines, blue for cytosines, and black for guanines) represent the four nucleotides.

See companion CD for color Fig. 5

other nucleotides will not adhere to the dyed nucleotides. Thus the enzymatic reaction generates DNA fragments of varying lengths that terminate at fluourescently labeled A, T, C, or G nucleotides. An automated sequencing machine then determines the underlying sequence for the range of DNA fragments created in the chemical reaction (*see* Fig. 5). The fluorescently tagged bases at the ends of the fragments are detected with a laser and a computer collects the resulting information. The order in which the particular tagged nucleotides are read reflects their order on the stretch of DNA that has been replicated. Each reaction reveals the sequence of at least 500 letters (G, A, T, C) of DNA before the process runs its course.

Once these relatively tiny sequences are obtained, their place in the overall genome DNA sequence must be determined. To achieve a working DNA sequence draft of the genome, two approaches were followed. The HGP began by creating detailed genetic and physical maps, to provide a framework for ordering the generated DNA sequences. Using this approach, the HGP divided the genome into about 30,000 segments (a technique called physical mapping), each containing an average of 100,000–200,000 base pairs. Each of these sections was then broken down into even smaller fragments, of about 2000 base pairs and sequenced. Initially, the plan was to put the fragments in order and systematically determine the sequence of each fragment so that the entire human DNA sequence would be revealed. This method produces a highly accurate sequence with few gaps. However, the up-front process of building the sequence maps is costly, time-consuming, and therefore, determines the speed at which the project is completed.

A second approach used by the HGP to generate a draft sequence of the human genome is what is called the whole genome shotgun or WGS (an approach that bypasses the need to construct physical maps). In WGS, sufficient DNA sequencing is performed at random so that each nucleotide of DNA in the genome is covered numerous times, in fragments of about 500 base pairs. Determining where those individual fragments fit in the overall DNA is accomplished through the use of powerful computers that analyze the raw data to find overlaps. A working draft of DNA sequence usually covers 95% of the genome (maintaining 99% accuracy) but it is divided into many unordered gapped segments. Additional sequencing is required to generate the finished DNA sequence, such that there are no gaps or ambiguities. The final sequence has an accuracy of greater than 99.99%. Only partial data was collected from each DNA fragment. This was then assembled to generate a working or rough draft. This change of strategy was partly due to the launching of a privately funded company, Celera Corporation.

Positional Cloning

Positional cloning is the process by which disease-causing genes are identified on the basis of their chromosomal location, with limited or no prior knowledge of the gene's function (*see* Fig. 6). Positional cloning can be divided into the following three steps.

Step 1: Family Studies

The first step in the positional cloning process is the collection of information on families who have the disease. Family trees (pedigrees) are established and DNA from blood is used for genetic analyses. Critical to this step is the diagnosis and assignment of proper phenotypic features to family members affected by the disease. Finding suitable families can be rate-limiting, particularly when the disease is rare, or for disorders in which affected individuals die at a young age. Two general approaches are used when collecting families for studies. A small number of very large families can be studied, where all affected members of the pedigree are known to have the same genetic disease, presumably caused by a mutation in a single gene. The alternative approach is to collect a large number of smaller families. This is easier to do for relatively common diseases, but it carries the risk that not all families may have the genetically identical disorder. However, with proper epidemiological studies, the mode of inheritance of the disease (X-linked, autosomal recessive, autosomal dominant or multifactorial) can often be determined.

Step 2: Mapping and Sequencing

The next step in positional cloning is to identify informative chromosomal rearrangements or genetic markers that are always found in family members affected by the disease. In the simplest case, the disease gene might be closely associated with a chromosomal anomaly, which helps define the position of the causative gene on the chromosome. Unfortunately this is a rare event and not the case for most genetic disorders. In the remainder of cases, genetic mapping is performed to determine the chromosomal region containing the disease gene being sought. This is accomplished by examining genetic markers that have already been mapped to particular chromosomes across the genome, to determine which ones are linked to the disease. If each family member that has the disease also has a particular DNA marker, it is likely that the gene responsible for the disease is in close proximity to that marker, thus defining where the gene search should be focussed.

Step 3: Candidate Gene Isolation and Mutation Analysis

The last step in positional cloning is to identify the genes within the candidate region. Historically, mapping the disease gene was followed by the construction of physical maps by ordering overlapping fragments of DNA along the region of interest and determining the nucleotide composition of the clones. Now that most of the human genome has been sequenced, candidate genes from the chromosomal region of interest are usually available for further study. However, a complete gene catalogue is not yet established. Therefore, at times, one must scan the DNA sequence to look for features characteristic of genes. Since the coding sequence of genes is usually not continuous, this is not always simple. Looking for exon-intron boundaries of coding and noncoding regions, amino acid encoding DNA sequences

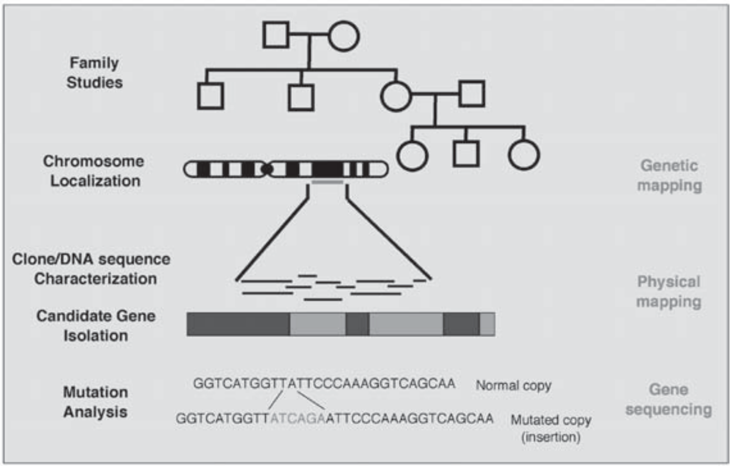


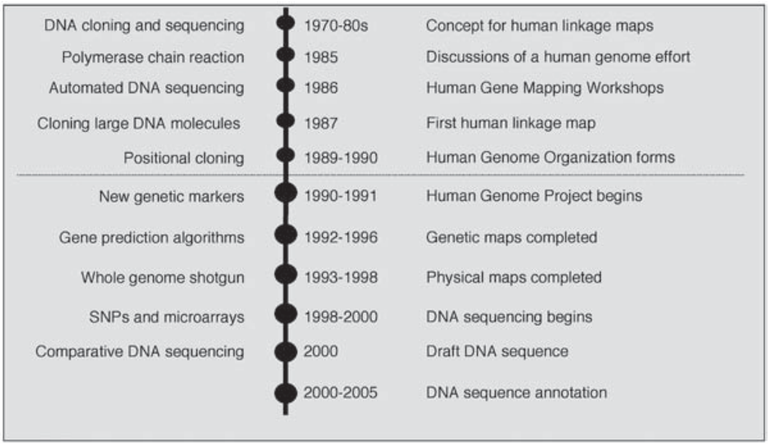
Fig. 6. Disease gene identification by positional cloning using mapping and DNA sequencing techniques.

to determine the start and end of the gene, or evolutionarily conserved DNA segments, all help to focus this search. The confirmation that a candidate gene is causative in the disease requires a direct association between a mutation in that gene and expression of the disease phenotype. The identified mutation should be present only in affected individuals and not in unaffected relatives or controls. However, in some individuals the mutation may not cause the disease, a phenomenon known as incomplete penetrance. The mutation could also cause a variation of the disease, a phenomenon known as reduced penetrance. In the case of a late-onset disorder like Huntington's disease, the mutation may not have had time to manifest itself. A key issue while looking for mutations in a disease gene is the ability to discriminate between nondisease-causing sequence polymorphisms that may just be linked to the disease gene and the actual disease-causing mutations. Various bioassays can be designed to show that a particular gene defect (correlated to an altered amino acid in the protein) can cause the phenotype in question. For example, the abnormal gene can be introduced in an animal model to see if it causes the disease. Alternatively, the normal form of the gene can be introduced and tested to determine if it can replace the abnormal copy of the gene.

The HGP is rapidly identifying all of the genes in the human genome. The goal of this effort is to find the location and pinpoint the function of the at least 30,000–40,000 human genes by the year 2005 (*see* Fig. 7). Positional cloning has thus far been used to identify hundreds of disease genes, including the gene for cystic fibrosis, Huntington's disease, some types of Alzheimer's disease, and early-onset breast cancer. The identification of a gene prior to the HGP is described in the following.

Identification of the Gene Causing Cystic Fibrosis

The gene for cystic fibrosis (CF) was identified in 1989, providing a paradigm for dozens of future disease gene discoveries using HGP data. CF is the most common fatal, autosomal recessive disorder in the Caucasian population. It is characterized by chronic



DNA cloning and sequencing	1970-80s	Concept for human linkage maps
Polymerase chain reaction	1985	Discussions of a human genome effort
Automated DNA sequencing	1986	Human Gene Mapping Workshops
Cloning large DNA molecules	1987	First human linkage map
Positional cloning	1989-1990	Human Genome Organization forms
New genetic markers	1990-1991	Human Genome Project begins
Gene prediction algorithms	1992-1996	Genetic maps completed
Whole genome shotgun	1993-1998	Physical maps completed
SNPs and microarrays	1998-2000	DNA sequencing begins
Comparative DNA sequencing	2000	Draft DNA sequence
	2000-2005	DNA sequence annotation

Fig. 7. Timeline of the Human Genome Project (HGP). A historical summary of some of the enabling technologies (on the left) and the achievements (on the right) leading up to and including the HGP are shown. The formal international HGP began in 1990. Other aspects of the HGP are the study of model organisms, analysis of genome variation, and the development of bioinformatics. Establishing training and public education programs, as well as the study of the ethical, legal, and social issues of genetics research in society, are also important priorities.

lung disease and pancreatic insufficiency, and affects 1 in 2500 individuals. The severity of the disease can often be correlated with the mutations present (*see* Fig. 4).

In order to locate the CF gene, researchers tested DNA from CF families with a large number of genetic markers. A total of 211 families were pooled and analyzed and eventually two markers located on chromosome 7 were found to flank the gene. This key finding, published in 1985, indicated that the CF gene must be inherited along with the markers on chromosome 7. Next, a combination of genetic techniques, including chromosome *walking* and *jumping*, was used. In walking to either direction of a known marker, an initial clone containing the marker gene is used to isolate another clone that contains overlapping information from the genome. The DNA segments can then be placed in an order corresponding to that on the chromosome and the process is repeated. A complimentary approach to this technique, called chromosome jumping, utilizes only the ends of cloned segments, making jumps over uninformative or repetitive regions of DNA. In the search for the CF gene, each DNA fragment isolated was compared to DNA from animal species in order to determine whether any of these fragments were conserved during evolution. A match was found with a sequence from chicken, mouse, and cow, suggesting that this gene was also encoded in these animals. This fragment of DNA represented the start of the CF gene and was used to identify the remainder of the gene. Since no chromosomal abnormality was evident in the patients, researchers then began searching for a difference between DNA from normal and CF patients. In 1989, a small deletion was found in a particular DNA fragment that appeared in 70% of the chromosomes from CF patients, but was absent from normal chromosomes. The gene was found to be 250 kb long, comprised of 24 exons, encoding a trans-membrane protein, 1480 aa long. This gene located on chromosome 7 was called CFTR (cystic fibrosis trans-membrane conductance regulator), which encodes a chloride channel protein. In the majority of patients, the error that causes CF is minute, only three of the base pairs are missing. This principal mutation, DeltaF508 (deletion of phenylalanine amino acid at position 508 in the protein), is found in approx 70% of carriers of European ancestry, and is enough to radically disrupt the function of patients' lungs, sweat glands, and pancreas. The CF gene is also associated with mutational heterogeneity. Over 550 other mutations have been identified. Many of these are extremely rare, but a few reach frequencies of 1–3 % in CF carriers.

The Human Genome Project and Disease Gene Identification

With the development of new mapping resources and technologies, and massive amounts of DNA sequence generated by the HGP, the ability to clone rearrangement breakpoints and map disease genes has been greatly simplified. This has also accelerated the pace of discovery of new disease loci and the underlying mutational mechanisms. For example, the gene for Parkinson's disease (alpha synuclein) on chromosome 4q21-q23 and the gene for speech and language disorder (FOXP2) on chromosome 7q31, were identified within only a few months of determining their chromosomal location.

After a decade of experience in positional cloning, and with the HGP DNA sequence now well-advanced, it has become possible to dissect the molecular genetics of multifactorial diseases such as cancer and cardiovascular disease. These involve multiple combinations of genes and strong environmental components. Scientists will continue to work on the HGP with an emphasis on annotating the DNA sequence to find new genes, determine the function of the gene products (functional genomics), and apply all of this information to the study of common diseases.

Glossary and Abbreviations

Acrocentric Human chromosomes (13, 14, 15, 21 and 22) with the centromere near one end.

Allele One of the alternative versions of a gene that may occupy a given locus.

Autosome The nuclear chromosomes other than the sex chromosomes.

Base pair (bp) A pair of complementary nucleotide bases in double stranded DNA.

CF Cystic fibrosis.

CFTR Cystic fibrosis trans-membrane conductance regulator.

Chromosome The threadlike structures in the cell nucleus consisting of chromatin that carries the genetic information (DNA).

Deletion A loss of a sequence of DNA from a chromosome.

DNA Deoxyribonucleic acid. The molecule that encodes the genes responsible for the structure and function of living organisms and allows the transmission of genetic information from parents to offspring.

Dominant A trait is dominant if it is phenotypically expressed in heterozygotes.

Exon A transcribed region of a gene that is present in mature messenger RNA.

Gamete A reproductive cell (ovum or sperm) with the haploid chromosome number.

Gene A hereditary unit, a sequence of chromosomal DNA that is required for the production of a functional product.

Genetic disorder A defect caused by genes.

Genetic map The relative positions of genes on the chromosomes.

Genetic marker A characteristic of DNA that allows different versions of a locus to be distinguished from one another and followed through families in genetic studies.

Genome The complete DNA sequence containing the entire genetic information.

Genotype The genetic constitution of an individual.

Germline The cell line from which the gametes are derived.

Haploid The chromosome number of a normal gamete.

HGP Human genome project.

Housekeeping genes Genes expressed in most or all cells because their products provide basic functions.

Insertion A chromosomal abnormality in which a DNA segment from one chromosome is inserted into a non-homologous chromosome.

Intron A segment of a gene that is initially transcribed but then removed from within the primary RNA transcript by splicing together the exon sequences on either side.

Locus The position occupied by a gene on a chromosome.

Mitochondrial DNA The DNA in the circular chromosome of mitochondria that is maternally inherited.

mRNA Messenger RNA.

Mutation Any permanent heritable change in the sequence of genomic DNA.

Nucleotide A molecule composed of a nitrogenous base, a 5-carbon sugar and a phosphate group.

Pedigree A diagram of a family history indicating the family members, their relationship with the proband and their status with respect to a particular hereditary condition.

Penetrance The fraction of individuals with a genotype known to cause a disease who have any signs or symptoms of the disease.

Phenotype The observed characteristics of an individual as determined by his or her genotype.

Physical map A map showing the order of genes along a chromosome and their distances apart in units such as base pairs.

Pleiotropy Multiple phenotypic effects of a single gene or gene pair.

Point mutation A single nucleotide base-pair change in DNA.

Polygenic Inheritance determined by many genes at different loci.

Polymorphism The occurrence in a population of two or more alternative genotypes.

Positional cloning The molecular cloning of a gene on the basis of its map position, without prior knowledge of the gene product.

Promoter A DNA sequence located in the 5' end of a gene at which transcription is initiated.

Recessive A trait that is expressed only in homozygotes or hemizygotes.

RNA Ribonucleic acid. A nucleic acid formed upon a DNA template, containing ribose instead of deoxyribose. Messenger RNA is the template on which polypeptides are synthesized.

Sequence The order of nucleotides in a segment of DNA or RNA.

Sex chromosomes The X and Y chromosomes.

SNP Single nucleotide polymorphism.

Somatic mutation A mutation occurring in a somatic cell.

STR Short tandem repeat.

Transcription The synthesis of a single-stranded RNA molecule from a DNA template in the cell nucleus, catalyzed by RNA polymerase.

Translocation The transfer of a segment of one chromosome to another chromosome.

Trisomy The state of having three representatives of a given chromosome instead of the usual pair.

VNTR Variable number tandem repeat.

WGS Whole genome shotgun.

Suggested Reading

Botstein, D., White, R. L., Skolnick, M., and Davis, R. W. (1980) Construction of a genetic linkage map in man using restriction fragment length polymorphisms, Am. J. Hum. Genet. 32, 314–331.

Chumakov, I. M., Rigault, P., Le Gall, I., Ballanne-Chantelot, C., Billault, A., Guillon, S., et al. (1995) A YAC contig map of the human genome, Nature 377 (Suppl), 175–297.

Cohen, D., Chumakov, I., and Weissenbach, J. (1993) A first generation physical map of the human genome, Nature 366, 698–701.

Collins, F. S. (1992) Positional cloning: Let's not call it reverse anymore, Nat. Genet. 1, 3–6.

Deloukas, P., Schuler, G. D., Gyapay, G., Beasley, E. M., Soderlund, C., Rodriguez-Tome, P. et al. (1998) A physical map of 30,000 human genes, Science 282, 744–746.

Dib, C., Faure, S., Fizames, C., Samson, D., Drouot, N., Vignal, A., et al. (1996) A comprehensive genetic map of the human genome based on 5,264 microsatellites, Nature 380, 152–154.

Donis-Keller, H., Green, P., Helms, C., Cartinhour, S., Weifenbach, B., Stephens, K., et al. (1987) A genetic linkage map of the human genome, Cell 51, 319–337.

Dunham, I. N., Shimizu, N., Roe, B., Chissoe, S., Hunt, A. R., Collins, J. E., et al. (2000), The DNA sequence of human chromosome 22, Nature 404, 904–920.

Hudson, T. J., Stein, L. D., Gerety, S. S., Ma, J., Castle, A. B., Silva, J., et al. (1995) An STS-based map of the human genome, Science 270, 1945–1954.

International Human Genome Sequencing Consortium (2001) Initial sequencing and analysis of the human genome, Nature 409, 860–914.

Kerem, B., Rommens, J. M., Buchanan, J. A., Markiewicz, D., Cox, T. K., Chakravarti, A., Buchwald, M., and Tsui, L-C. (1989) Identification of the cystic fibrosis gene: Genetic analysis, Science 245, 1073–1080.

Riordan, J. R., Rommens, J., Kerem, B., Alon, N., Rozmahel, R.,Grzelczak, Z., et al. (1989) Identification of the cystic fibrosis gene: Cloning and characterization of complementary DNA, Science 245, 1066–1073.

Rommens, J. M., Iannuzzi, M. C., Kerem, B., Drumm, M. L., Melmer, G., Dean, M., et al. (1989) Identification of the cystic fibrosis gene: chromosome walking and jumping, Science 245, 1059–1065.

The Huntington's Disease Research Collaborative Group (1993) A novel gene containing a trinucleotide repeat is expanded and unstable on Huntington's disease chromosomes, Cell 72, 971–983.

Tsui, L-C. and Estivill, X. (1991) Identification of disease genes on the basis of chromosomal localization, Genome Anal. 3, 1–36.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., et al. (2001) The sequence of the human genome, Science 291, 1304–1351.

Zielenski, J. and Tsui, L. C. (1995) Cystic fibrosis: Genotypic and phenotypic variations, Annu. Rev. Genet. 29, 777–807.

B. Clinical Human Genetics

10. Heredity
11. The Clinical Genetics Databases
12. Population Genetics

10 Heredity

C. A. Rupar

Introduction

Clinical geneticists classify human traits and genetic disorders on the basis of the patterns of inheritance in families: sporadic, autosomal dominant, autosomal recessive, X-linked, and mitochondrial. A compendium of genetic information, McKusick's Online Mendelian Inheritance in Man (OMIM) (*see* Website: http://www.ncbi.nlm.nih.gov/omim/), has been assembled primarily based on clinical observations as to how traits and disease states appear to be inherited.

Identification of the patterns of inheritance of genetic disorders has enabled geneticists to make significant contributions to health care. The recognition of the pattern of inheritance of a disorder in a family may be an important diagnostic aid, permit the identification of individuals in a family who are at risk to develop the disorder, and enable informed counseling of family members about the recurrence risks of having another affected child.

Early in 2001, the results of the publicly and privately funded human genome sequencing projects indicated that the human genome contains about 3 billion bases arranged in a genome, which encodes at least 35,000 genes. This represents considerably fewer genes than had been anticipated. As of September 22, 2001 the definitive catalog of human genes, OMIM listed a total of 12,954 loci with 7,441 mapped to specific sites in the genome. Clearly there are many more genes to be identified and the knowledge of how the expression of genes are regulated spatially and temporally is still in its infancy.

The majority of clinical genetics is devoted to the family, but genes are shared not only within families but also within populations. Many rare genetic traits affect individuals throughout the world, whereas other normally rare traits are much more prevalent in some populations. There are a number of reasons why a population may have a higher frequency of a particular allele that may be favorable or unfavorable. Individuals in populations whose gene pool has been relatively isolated for generations and populations whose gene pools originate from a small number of founder individuals are more likely to share alleles that may be rare outside that population.

In other situations one population may experience selective pressures that favour a particular allele but another population may not experience the same selective pressure. Perhaps the best example is heterozygosity, when an individual carries both alleles, for otherwise deleterious globin gene mutations it protects individuals in tropical climates against malaria.

Dominance and Recessiveness

Dominance and recessiveness refer to clinical phenotypes. A condition is recessive when the phenotype of a heterozygous individual is the same as the phenotype of homozygous normal individuals. However, upon closer inspection, heterozygous individuals in many recessive conditions may have a measurable biochemical difference when compared to homozygous normal individuals. For example, phenylketonuria (PKU) is most often due to mutations in the *PAH* locus, which codes for phenylalanine hydroxylase. Individuals with PKU are unable to convert phenylalanine to tyrosine and have plasma concentrations of phenylalanine much higher than normal individuals. Individuals who are heterozygous for PKU do not have a clinical phenotype but when plasma concentrations of phenylalanine and tyrosine are measured, heterozygotes may be distinguished from homozygous normal individuals.

In principle a condition is dominant when the clinical phenotype of heterozygous individuals is indistinguishable from homozygous affected individuals. There are probably few examples of true dominant disorders, with the most notable example being Huntington's disease. Most mutations are deleterious when compared to the wild-type or *normal* allele and behave in a recessive manner. It is unlikely that a newly and randomly introduced change in protein structure will result in a protein with improved function.

The explanation as to why a given disorder is inherited in a dominant or recessive manner is often not understood but there are observations that do provide clues. For example most enzyme deficiencies are inherited in a recessive manner with the heterozygous individual not having a phenotype. Presumably most enzymes are expressed in sufficient excess that a heterozygote who may have about half (to a first approximation) of the enzyme activity of a homozygous normal individual has a level of activity that prevents a phenotype.

Dominantly inherited disorders are often caused by genes that code for structural or developmentally important proteins where presumably half of the normal amounts of the gene product are inadequate. There are also examples of dominant mutations in regulatory enzymes. Porphobilinogen deaminase (PBG) is a notable example. PBG catalyses the first and rate-limiting step in heme biosynthesis. A deficiency in PBG deaminase results in the autosomal dominantly inherited condition, acute intermittent porphyria. Another type of example of dominant inheritance is familial hypercholesterolemia, owing to a deficiency of the low-density lipoprotein (LDL) receptor. This plays a key role in regulating cellular cholesterol metabolism.

Whether a given mutation behaves in a dominant or recessive manner is best explained by knowledge of the biochemistry and physiology of the gene product of specific mutations. Indeed, different mutations in the same gene may behave in dominant or recessive manners. An important concept is that of dominant negativeness, which describes the situation that occurs when a mutant protein from one chromosome interferes with the function of the normal protein from the other chromosome. There are elegant examples of dominant negativenss in disorders of the structural protein collagen and in some multimeric enzymes.

There are also examples of dominantly inherited mutations that result in a gain of function for a protein. This new function may be deleterious or even toxic to cells. An example is Huntington's disease, where there is a gain of function associated with the expansion of a CAG trinucleotide repeat. Other examples like hypermethioninema described below, cause loss of function.

Methionine Adenosyl Transferase OMIM #250850

Methionine adenosyl transferase (MAT) catalyses the synthesis of adenosylmethionine from methionine and ATP. Adenosylmethionine is an important methyl donor in transmethylation reactions and a deficiency of MAT results in hypermethioninemia. Most often the phenotype, hypermethioninemia, is inherited in an autosomal recessive manner with at least 17 mutations in *MAT1A* identified in the human population. However there are families in which the phenotype is inherited in an autosomal dominant manner. MAT exists in two forms MATI and MAT III. MATI and III are, respectively, homotetramers and homodimers of the peptide that is encoded by the *MATA1* gene. A mutation that results in the substitution of arginine with histidine at position 264 greatly reduced MATI/III activity by preventing homodimer formation that is essential for catalytic activity and is inherited in a dominant manner.

Understanding Pedigrees

Pedigrees are an important tool that geneticists use to understand the pattern of inheritance of a disorder or trait in a family. Pedigrees are drawn with standardized symbols and follow some simple conventions. Each generation is assigned a Roman numeral with the first or oldest generation of the pedigree numbered I. Each individual within a generation is assigned a number with number 1 being the individual at the left of the pedigree. Thus every individual in a pedigree has a unique number based on generation number and position within the generation. Within a family, the children are drawn in decreasing age from left to right. Figure 1 displays the various symbols used in the construction of pedigrees.

Autosomal Dominant Inheritance

An individual with an autosomal dominant condition has a mutation in only one allele, with the allele on the other chromosome being normal. Families with autosomal dominant disorders typically show vertical patterns of inheritance in pedigrees. The low frequency of most alleles that contain mutations makes the likelihood of an individual to inherit dominant mutant alleles from both parents quite low, but nevertheless, this has been observed. Typically individuals who are *double dominant* will have a more severe phenotype.

In autosomal dominantly inherited disorders a mutant gene may not always be expressed in a phenotype or the phenotype that is expressed may be variable in different individuals even in the same family. These concepts are referred to as reduced penetrance and variable expressivity. The underlying biological explanations of reduced penetrance and variable expressivity are generally not understood but presumably lies in the effects of other loci or environmental factors on the expression of the phenotype.

Often in the family of an affected individual there is no evidence of the previous presence of a dominant gene. This may be explained by the mutation in the affected

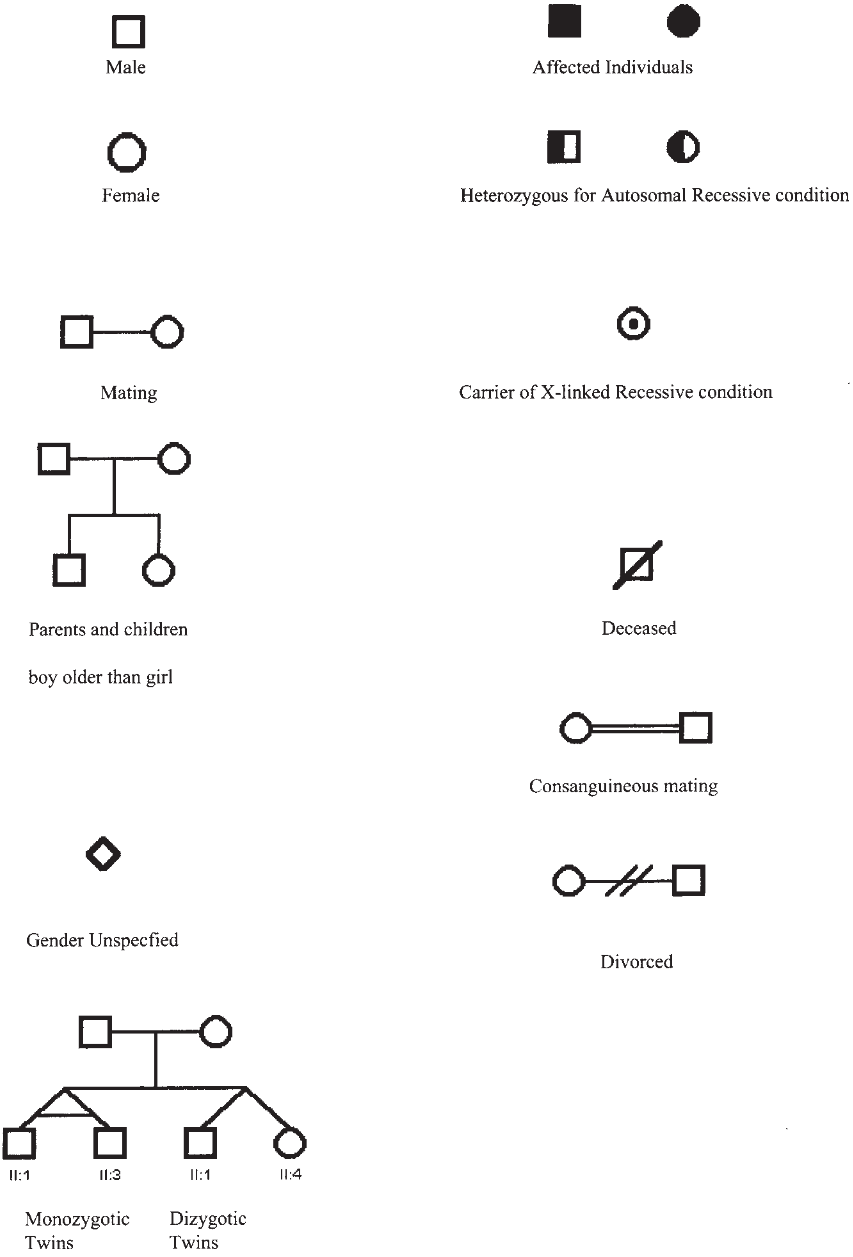


Fig. 1. The key to understanding pedigrees.

individual being a new mutation. New mutation rates have been determined for a number of disorders and typically occur in the range of 5 × 10^{-6} to 1 × 10^{-4} mutations per gamete per generation. Because a new mutation is a relatively rare event, it is unlikely to happen twice with the same couple. Empirical data gained by studying many families indicate that the recurrence risk of having another affected child is usually significantly greater than that of acquiring a new mutation. Those couples who do have another affected child are most easily explained by one of the parents being a *gonadal mosaic*. That is, a mutation has occurred in a germ-cell lineage and that parent may have a mixed population of sperm or ova with some cells having the normal allele and others having the mutation.

Trinucleotide repeat disorders are a particularly interesting group of dominantly inherited disorders. The underlying mutation is the expansion of a repetitious region of the gene. There are several triplet expansion disorders with many being primarily neurological. The following examines the clinical and genetic phenotype of muscular dystrophy.

Myotonic Dystrophy (OMIM # 160900)

Myotonic dystrophy is a fascinating example of a trinucleotide repeat as a mutation. Individuals with myotonic dystrophy have a progressive state of muscle tightness due to chronic spasms and other clinical features, including cataracts, early frontal hair loss, electrocardiographic changes, muscle wasting, and hypogonadism. Clinical geneticists have recognized for years the phenomena of *anticipation* in myotonic dystrophy: each successive generation has a more severe and earlier presenting disease. The gene for myotonic dystrophy is located at chromosome position 19q13.3 and codes for the protein dystrophia myotonia protein kinase. In the 3' untranslated region of the mRNA the sequence CTG is repeated from 5–35 times in tandem in unaffected individuals. In individuals with myotonic dystrophy, the repeat has expanded and may become as long as 5 kb. There is a rough correlation between the length of the repeat and the severity of disease. The repeat often becomes longer in successive generations although the opposite has also been observed. Often, the most severely affected individuals, those with congenital myotonic dystrophy, inherit the mutation from their mothers.

Characteristics of Autosomal Dominant Inheritance

- The trait often appears in every generation. Exceptions occur in the case of new mutations and with disorders that can have reduced penetrance or expressivity. Reduced penetrance occurs when some individuals who have inherited a particular geneotype do not express the typical phenotype. Penetrance can be described mathematically as the percentage of individuals who have the gene for a condition who actually show the trait. Expressivity of a phenotype is variable when different individuals with the same mutated gene express different phenotypes.
- Affected individuals will pass on the mutation to half of their children.
- Males and females are equally likely to inherit the trait.
- Unaffected members of the family will not pass on the trait to their children.

A typical pedigree for an autosomal dominant inherited disorder is shown in Fig. 2.

Autosomal Recessive Inheritance

An autosomal recessive trait is only expressed in homozygotes. Affected individuals have inherited a gene with a mutation from both parents. The parents, who are heterozygotes, are generally unaffected. Most inborn errors of metabolism, which are due to enzyme deficiencies, are inherited in an autosomal recessive manner. Recessive disorders are rare in most populations. However there are many groups where certain rare recessive disorders are relatively common because the population has been isolated by

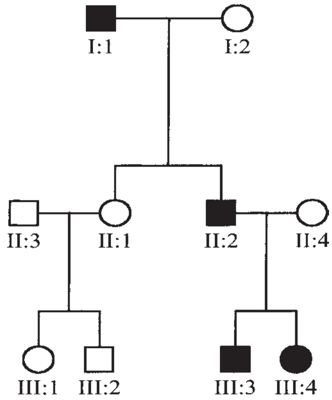


Fig. 2. An autosomal dominant pedigree.

geographical, cultural, religious, or linguistic barriers. Another situation that raises the risk for recessive disorders is when the parents are related, which is termed consanguinity. The more closely the parents are related, the greater the risk of sharing rare recessive alleles. In the extreme cases of parent-child and brother-sister matings, the children are homozygous at one quarter of their gene loci.

There are often many mutations in the population and it is of considerable interest to geneticists to establish genotype-phenotype correlations. Can the knowledge of an individual's genotype at a locus be used to predict the severity of disease? This becomes complex in autosomal recessive disorders where many affected individuals are compound heterozygotes, i.e., they inherited different mutations from each parent. Phenylketonuria, described below, is a classic example of an autosomal recessive disorder.

Phenylketonuria (OMIM # 261600)

Phenylketonuria is a classic example of an autosomal recessive disorder. Most individuals with phenylketonuria have mutations in the gene, which codes for phenylalanine hydroxylase (PAH). *PAH* converts phenylalanine to tyrosine and when it is dysfunctional the plasma and tissue concentrations of phenylalanine rise dramatically. This causes a devastating progressive neurological disorder that results in profound developmental delay with affected individuals dependent on others for their care. The incidence of PKU is about 1 in 15,000 births, which means nearly 1 in 60 in the population is a heterozygote carrying one *PAH* allele with a mutation. Many countries throughout the world screen newborn children for PKU because the profound developmental delay is preventable by a diet that is low in phenylalanine. PKU, like many other genetic disorders has a dedicated website. This provides information for families and a database for students and researchers (*see* Website: http://ww2.mcgill.ca/pahdb). Some mutations in the *PAH* locus have a relatively small impact on the function of phenylalanine hydroxylase and result in benign hyperphenylalaninemia, whereas other mutations have a severe impact on the presence or function of the enzyme and result in PKU. The presence of the phenotype is determined by the concentration of plasma phenylalanine. Benign hyperphenylalaninemia does not require dietary treatment.

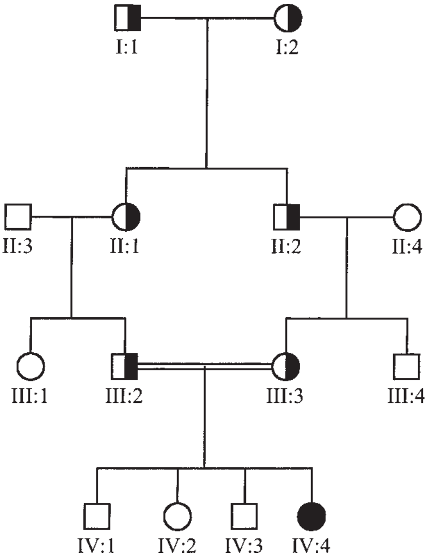


Fig. 3. An autosomal recessive pedigree.

Characteristics of Autosomal Recessive Inheritance

- Usually the phenotype is present in brothers and sisters not in parents.
- Statistically one-quarter of the sibs of an affected child will also be affected.
- Males and females are equally likely to be affected.
- Occurs more frequently in consanguineous relationships.

A typical pedigree for an autosomal recessive disorder is shown in Fig. 3.

X-linked Inheritance

Phenotypes of mutations in genes on the X chromosome are inherited in characteristic patterns that are determined by males having one X chromosome and females having two X chromosomes. Most disorders on the X chromosome are recessive in nature and are expressed primarily in males. Females are not always totally protected by the X chromosome with the normal allele because the Lyon hypothesis predicts random inactivation of one of the X-chromosomes early in embryonic development. Females with a mutation on one X chromosome then become mosaics with some cells inactivating the chromosome with the normal sequence gene, while others inactivate the X chromosome with the mutation. By random inactivation, some females inactivate a high proportion of the normal X chromosome. Even so, they essentially have a milder phenotype of the male disorder. In some disorders, a female may not have a phenotype, she may have a measurable biochemical abnormality as typified by X-inactivation of ornithine transcarbamylase as X described next.

Ornithine Transcarbamylase (OMIM # 311250)

Ornithine transcarbamylase catalyzes the synthesis of citrulline from ornithine and carbamyl phosphate in an early step in the urea cycle. The most common presentation

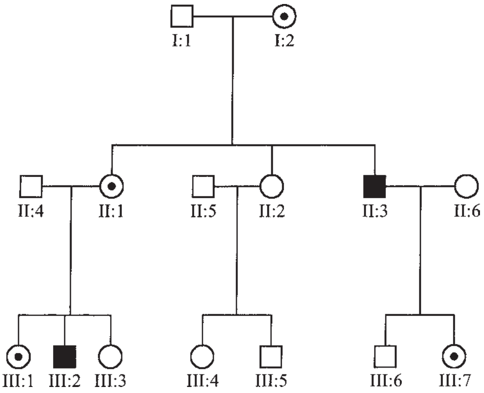


Fig. 4. An X-rated pedigree.

of affected males is an acute life-threatening hepatic encephalopathy brought on by very high levels of blood ammonium at a few days of age. Affected females usually present later in infancy or childhood with a less severe disease. These females have inactivated too great of a proportion of the X chromosomes carrying the normal allele in their livers. Like many X-linked disorders, many different mutations have occurred in the population. When parents of children with ornithine transcarbamylase deficiency are genotyped, it is apparent that frequency of new mutations in sperm is much higher than ova.

Characteristics of X-Linked Inheritance

- The incidence of the disorder is much greater in males than females.
- All the daughters of an affected male will be carriers of the mutant allele.
- A father can never transmit the mutant allele to his sons.
- Female carriers may show variable expression of the disorder.

A typical pedigree for an X-linked trait is shown in Fig. 4.

Mitochondrial Inheritance

Mitochondria, the powerhouse of the cell, are unique structures possessing their own DNA. The complete sequence of human mitochondrial DNA has been known since the early 1980s. It is a 16,569 nucleotide double-stranded molecule that codes for 13 polypeptides essential for oxidative phosphorylation, 22 tRNAs and 2 ribosomal RNAs. However, the vast majority of mitochondrial proteins are encoded in the nucleus. There has been intense interest in the contribution of mitochondria DNA to human disorders (*see* Website: http://www.gen.emory.edu/mitomap.html). This has enabled the identification of mutations that cause genetic disorders. Approximately 125 pathological point mutations have been identified in the mitochondrial DNA sequence. There have also been a large number of deletions characterized in patients with mitochondria DNA disorders. Distinguishing pathological mutations from benign polymorphisms is sometimes difficult. Unlike the nuclear encoded mutations, there are yet expression systems to study the effects of a mitochondrial DNA mutation on protein or cellular function.

At the time of conception all mitochondria originate from the ovum; the sperm makes no contribution in humans. Thus, any attribute or disorder that is a consequence of the mitochondrial genome is inherited along the maternal lineage. A mitochondrion may contain several copies of its genome and many cell types have large numbers of mitochondria; consequently, mitochondrial DNA is present in multiple copies within a cell.

A mutation in mitochondrial DNA, either inherited or a new mutation, can result in a mixed population of mitochondrial DNA consisting of normal sequence DNA and DNA with the mutation. This is called heteroplasmy. During mitosis, mitochondria in the parent cell are distributed to the daughter cells in a random manner resulting in daughter cell populations that may have different populations of mitochondrial DNA. These range from cells having only the normal sequence cells to cells that have a higher percentage of mutant DNA than the parent cell. The maternal pattern of inheritance mitochondrial DNA mutations provides a major clue as to whether a mitochondrial disorder is due to a mutation in a nuclear encoded or mitochondrial-encoded gene.

Mitochondrial DNA accumulates spontaneous mutations about 10 times faster than nuclear DNA. Two possible reasons for this sensitivity to damage are the presence of free radicals in mitochondria as a consequence of the electron transport chain and the lesser competence of mitochondria when compared to the nucleus to repair damage to DNA. Mitochondrial DNA mutations exhibit a threshold effect, in that, often about 85% of the mitochondrial DNA must contain a mutation before a clinically detectable phenotype occurs. An example of MELAS syndrome is described as follows.

MELAS Syndrome (OMIM # 540000)

MELAS syndrome, which is the acronym of mitochondrial myopathy, encephalopathy, lactic acidosis, and stroke-like episodes, is most frequently due to an A to G mutation at position 3243 of the mitochondrial genome. This region of the mitochondria codes for a leucine tRNA. Interestingly, many individuals with the nt 3243 mutation do not have MELAS syndrome but rather develop seemingly unrelated health problems such as bilateral sensorineural hearing loss and adult-onset diabetes. There are other mutations in the same leucine tRNA that also cause MELAS, but also mutations that cause seemingly unrelated disorders such as cardiomyopathy. Genotype-phenotype correlations within mitochondria DNA disorders are not often evident, making molecular diagnoses difficult. Presumably much of the variability reflects random segregation at cell division and threshold effects.

Characteristics of Mitochondrial Inheritance

- Inheritance is only along the maternal lineage. A male cannot pass the disorder to his children.
- There is no gender preference.
- Many, but not necessarily all, of an affected female's children will inherit the mutation but the clinical impact may be quite variable.

A typical pedigree for a mitochondrial inherited trait is shown in Fig. 5.

Multifactorial Inheritance

Although there are thousands of disorders that are inherited along recognisable Mendelian patterns of inheritance, most are relatively uncommon. Many common diseases such as schizophrenia, diabetes, alcoholism, coronary artery disease, and neural tube defects often "run in families" but are not inherited in a Mendelian pattern. These

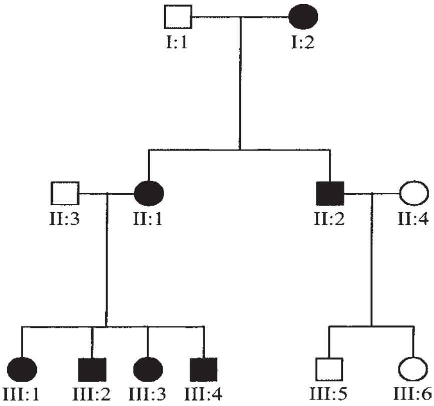


Fig. 5. Pedigree for a mitochondrial inherited trait.

common disorders are inherited in a multifactorial manner that includes the influences of multiple genetic and environmental factors. It will be one of the great accomplishments of human genetics to identify and quantify the contributions of genes to the risks of developing common disorders.

Studies using the incidence of disorder data in monozygotic (identical) and dizygotic (fraternal) twins, have made significant contributions to identifying the presence of a genetic susceptibility component to common diseases. As expected, disorders as diverse as cleft lip and palate, insulin-dependent diabetes, and coronary artery disease are more concordant in monozygotic than dizygotic twins.

In most cases, all the genetic and environmental contributions to the end point of a common disease have not yet been identified. However, there are models to help understand the principles of the roles of multiple contributions. The multiple additive locus model views phenotypes as the sum of the contribution of individual alleles. If at a given gene there are two possible alleles: a and b, then there are three possible genotypes: aa, bb, and ab. Alleles a and b may have different functional efficiencies and hence genotypes aa, bb, and ab may have quantitatively different contributions to a metabolic pathway. Applying the same logic to several genes whose gene product contributes to a phenotypic endpoint indicates how related individuals who share a number of alleles that may be less favorable, share an increased risk. It is easy to see how the multiple additive locus model can be applied to traits that are quantitative with endpoints that cover a continuum of values such as blood pressure.

Other multifactorial traits appear to be better explained by a threshold model of inheritance. The threshold model is particularly applicable to traits such as congenital malformations that tend to behave in an all-or-none manner. In the threshold model of multifactorial inheritance, when a combination of genetic and environmental factors reach a threshold value, the disorder is expressed. The threshold may be envisioned as having a constant value. The threshold may be exceeded in an individual by increasing either the genetic or environmental burden. Close relatives of an individual who exceeded the threshold for a congenital malformation, for example, share a greater percentage of genes than unrelated individuals and are more likely to exceed the threshold. A classic example of a multifactional genetic effect is typified by neural tube defects as described.

Neural Tube Defects

Neural tube defects in the absence of other congenital malformations are a heterogenous group of disorders that range from anencephaly to spina bifuda occulta, a benign form of spina bifuda. Neural tube defects are caused by the embryological failure of the neural groove to close and form the neural tube at about the third week after conception. The most familiar type of neural tube defect is spina bifuda in the lumbar region. This results in the tissues covering the spinal cord and often the spinal chord protrudes from the back. After surgical repair the infant will often be paralyzed below the hips. The world-wide incidence of neural tube defects is about 1 per 1000 live births with some populations such as in Great Britain having a much higher incidence. The genetic contributions to neural tube defects are illustrated by the fact that the occurrence of neural tube defects in first degree (parent or sibling) relatives is much greater than in the general population with a recurrence risk of about 3%. The environmental aspect to the etiology of neural tube defects is illustrated by the fact that the incidence of neural tube defects can be greatly reduced by supplementing the diet of women with folic acid prior to conception.

11 The Clinical Genetics Databases

Peter J. Bridge

Introduction

This introduction to Clinical Genetics reviews a selection of major internet resources drawn from the current large and growing listing. These include genetic variation, diagnostics, laboratory services, regulatory bodies and sequence data. Each of these sites can be regarded as the starting point for a journey to the wealth of information. Many sites provide a page of further links for those seeking additional or specific information. A selection of sites are shown in the figures and all referenced and linked websites are provided on the CD supplement.

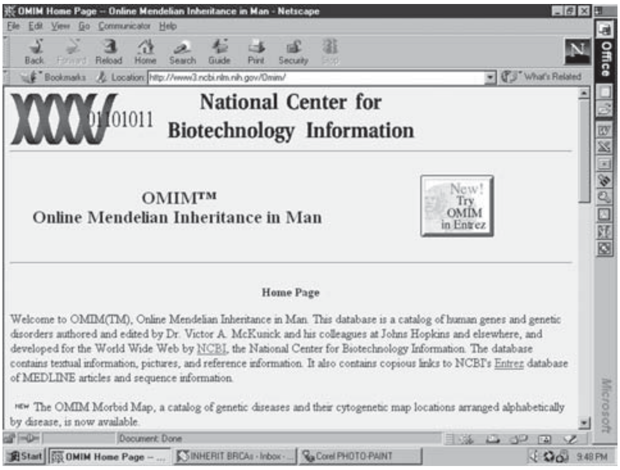
OMIM (On-line Mendelian Inheritance in Man)

The principal catalog of genetic variation in humans has long been McKusick's Mendelian Inheritance in Man (MIM). This has been published in book form and revised approx every 2–3 yr by the Johns Hopkins University Press since 1966. An internet-based database and catalog called On-line Mendelian Inheritance in Man (OMIM) has now become available at Website: http://www.ncbi.nlm.gov/omim as part of the NCBI Entrez System. This curated database is updated daily and contains direct links to most of the other important clinical genetics databases as well as comprehensive hypertext links to other entries within OMIM.

The scale of OMIM is staggering. Great detail is provided on all known human phenotypes, and more recently, on many genes that have yet to be associated with any specific phenotype. When using OMIM it is important to be aware that the entries frequently have a significant historical component, from which outdated material is slowly purged. It is often difficult to be absolutely clear, which parts are a historical description vs current consensus.

See companion CD for color Figs. 1 and 2

Figure 1 shows the top of the entry page. Statistics concerning the database and the conventions used within it must be accessed through this entry page. As shown in Fig. 2, one can also proceed directly to the "searchomim.html" page for routine genetic searches. Try entering **cystic fibrosis** in the search box and clicking **title** in the field box. Several entries will be returned. Entry 602421 contains the gene and 219700 is the entry for the clinical phenotype. Either entry will also provide clickable boxes that contain direct links to the specific entry on cystic fibrosis in other databases.


OMIM Home Page -- Online Mendelian Inheritance in Man - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www3.ncbi.nlm.nih.gov/Omim/
What's Related
01101011
New! Try OMIM in Entrez
Document: Done
Start OMIM Home Page -- ... INHERIT BRCAs - Inbox - ... Corel PHOTO-PAINT
9:48 PM
Office
Microsoft

National Center for Biotechnology Information

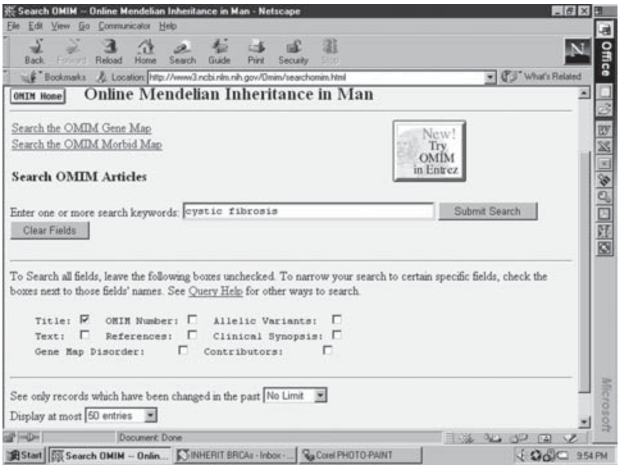
OMIM™
Online Mendelian Inheritance in Man

Home Page

Welcome to OMIM(TM), Online Mendelian Inheritance in Man. This database is a catalog of human genes and genetic disorders authored and edited by Dr. Victor A. McKusick and his colleagues at Johns Hopkins and elsewhere, and developed for the World Wide Web by NCBI, the National Center for Biotechnology Information. The database contains textual information, pictures, and reference information. It also contains copious links to NCBI's Entrez database of MEDLINE articles and sequence information.

NEW The OMIM Morbid Map, a catalog of genetic diseases and their cytogenetic map locations arranged alphabetically by disease, is now available.

Fig. 1. OMIM: Online Mendelian Inheritance in Man home page.


Search OMIM -- Online Mendelian Inheritance in Man - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www3.ncbi.nlm.nih.gov/Omim/searchomim.html
What's Related
OMIM Home
New! Try OMIM in Entrez
Submit Search
Clear Fields
Document: Done
Start Search OMIM -- Onlin... INHERIT BRCAs - Inbox - ... Corel PHOTO-PAINT
9:54 PM
Office
Microsoft

Online Mendelian Inheritance in Man

Search the OMIM Gene Map
Search the OMIM Morbid Map

Search OMIM Articles

Enter one or more search keywords: cystic fibrosis

To Search all fields, leave the following boxes unchecked. To narrow your search to certain specific fields, check the boxes next to those fields' names. See Query Help for other ways to search.

```
Title: ☑   OMIM Number: □   Allelic Variants: □
Text: □    References: □    Clinical Synopsis: □
Gene Map Disorder: □       Contributors: □
```

See only records which have been changed in the past: No Limit

Display at most: 50 entries

Fig. 2. Online Mendelian Inheritance in Man search page.

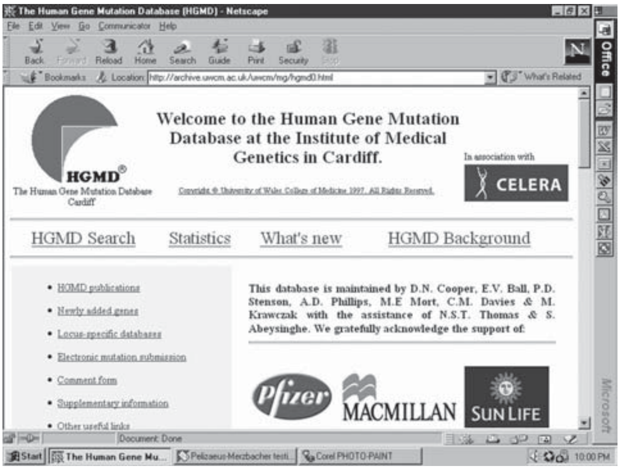
Databases of Human Mutations

Lists of mutations in each gene are compiled at the Human Gene Mutation Database (HGMD). This has a new web address (*see* Website: http://www.hgmd.org). Clicking on the **HGMD Search** link (top left, *see* Figure 3), permits progression to the search page where diseases or genes can be entered in a fashion similar to the OMIM search page. Remember that this site is based in the UK. Entering, for instance, a well-known disorder named hemochromatosis, returns nothing because in the UK it is spelled haemochromatosis. One might expect similar results from many other databases that are maintained outside North America like the Human Genome Variation Society (*see* Website: http://ariel.ucs.unimelb.edu.au:80/~cotton/mdi.htm), based at the University of Melbourne.

WWW

See companion CD for color Fig. 3

WWW



The Human Gene Mutation Database (HGMD) - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://archive.uwcm.ac.uk/uwcm/mg/hgmd0.html
What's Related
Office
HGMD®
The Human Gene Mutation Database Cardiff
In association with
CELERA
HGMD Search
Statistics
What's new
HGMD Background
Pfizer
MACMILLAN
SUN LIFE
Microsoft
Document: Done
Start
The Human Gene Mu...
Pelizaeus-Merzbacher testi...
Corel PHOTO-PAINT
10:00 PM

Welcome to the Human Gene Mutation Database at the Institute of Medical Genetics in Cardiff.

Copyright © University of Wales College of Medicine 1997. All Rights Reserved.

- HGMD publications
- Newly added genes
- Locus-specific databases
- Electronic mutation submission
- Comment form
- Supplementary information
- Other useful links

This database is maintained by D.N. Cooper, E.V. Ball, P.D. Stenson, A.D. Phillips, M.E Mort, C.M. Davies & M. Krawczak with the assistance of N.S.T. Thomas & S. Abeysinghe. We gratefully acknowledge the support of:

Fig. 3. Genetics Societies home page.

Gene-Specific Databases (Caution)

There are a great number of disease-specific mutation databases on the internet. Most are associated with research laboratories and it is difficult to judge the accuracy and completeness of many of these: user beware! A comprehensive listing of such sites is maintained at Cardiff (without warranty) by the Human Gene Mutation Database (*see* Website: http://archive.uwcm.ac.uk/uwcm/mg/docs/oth_mut.html). The Cardiff site also hosts a database called FIDD, which provides frequencies of inherited disorders around the world (*see* Website: http://archive.uwcm.ac.uk/uwcm/mg/fidd/).

Repositories of Sequence Data

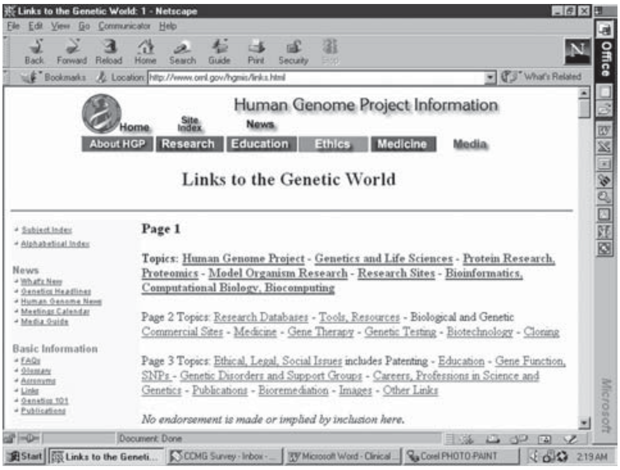
See companion CD for color Fig. 4

The cDNA sequence of most genes can be conveniently accessed through the Human Gene Mutation Database. Genomic sequences are found at several sites including the Genome Database (GDB; *see* Website: http://gdbwww.gdb.org/) and the National Center for Biotechnology Information (NCBI; *see* Website: http://www.ncbi.nlm.nih.gov/). GDB links are provided through OMIM. One of the more comprehensive starting points is a site on the Human Genome Project Information page maintained for the Department of Energy by the Oak Ridge National Laboratory (ORNL) (*see* Fig. 4, *see* Website: http://www.ornl.gov/hgmis/links.html).

GeneClinics

See companion CD for color Fig. 5

Each entry in the GeneTests database is provides diagnostic and clinical management information. It curated by an expert or panel of experts on that particular disease. Clicking the **Find Disease-Specific Information** heading at the top right of the entry page (*see* Fig. 5), presents a disclaimer page that the user must indicate he/she has read before proceeding. As an example, if the term *cystic* is entered into the search box, you will retrieve cystic fibrosis, polycystic kidney disease, and autosomal recessive. Clicking on the **text info** box next to cystic fibrosis will lead to a long and very detailed article on the clinical aspects of cystic fibrosis. This database is partnered with the GeneTests database.


Links to the Genetic World: 1 - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www.ornl.gov/hgmis/links.html
What's Related
Office
Microsoft
Document: Done
Start Links to the Geneti... CCMG Survey - Inbox - ... Microsoft Word - Clinical ... Corel PHOTO-PAINT 2:19 AM

Human Genome Project Information

Home Site Index News

About HGP Research Education Ethics Medicine Media

Links to the Genetic World

- Subject Index
- Alphabetical Index

News
- What's New
- Genetics Headlines
- Human Genome News
- Meetings Calendar
- Media Guide

Basic Information
- FAQs
- Glossary
- Acronyms
- Links
- Genetics 101
- Publications

Page 1

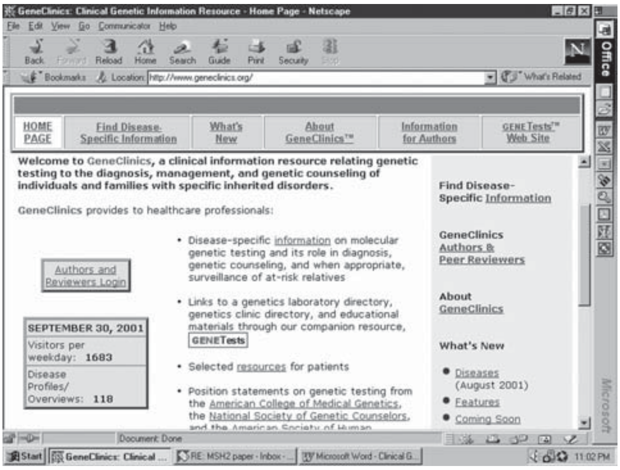
Topics: Human Genome Project - Genetics and Life Sciences - Protein Research, Proteomics - Model Organism Research - Research Sites - Bioinformatics, Computational Biology, Biocomputing

Page 2 Topics: Research Databases - Tools, Resources - Biological and Genetic Commercial Sites - Medicine - Gene Therapy - Genetic Testing - Biotechnology - Cloning

Page 3 Topics: Ethical, Legal, Social Issues includes Patenting - Education - Gene Function, SNPs - Genetic Disorders and Support Groups - Careers, Professions in Science and Genetics - Publications - Bioremediation - Images - Other Links

No endorsement is made or implied by inclusion here.

Fig. 4. Genetic Professional Societies home page.


GeneClinics: Clinical Genetic Information Resource - Home Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www.geneclinics.org/
What's Related
Office
Microsoft
Document: Done
Start GeneClinics: Clinical ... RE: MSH2 paper - Inbox - ... Microsoft Word - Clinical G... 11:02 PM

HOME PAGE | Find Disease-Specific Information | What's New | About GeneClinics™ | Information for Authors | GENETests™ Web Site

Welcome to GeneClinics, a clinical information resource relating genetic testing to the diagnosis, management, and genetic counseling of individuals and families with specific inherited disorders.

GeneClinics provides to healthcare professionals:

Authors and Reviewers Login

SEPTEMBER 30, 2001
Visitors per weekday: 1683
Disease Profiles/ Overviews: 118

- Disease-specific information on molecular genetic testing and its role in diagnosis, genetic counseling, and when appropriate, surveillance of at-risk relatives
- Links to a genetics laboratory directory, genetics clinic directory, and educational materials through our companion resource, GENETests
- Selected resources for patients
- Position statements on genetic testing from the American College of Medical Genetics, the National Society of Genetic Counselors, and the American Society of Human

Find Disease-Specific Information

GeneClinics Authors & Peer Reviewers

About GeneClinics

What's New

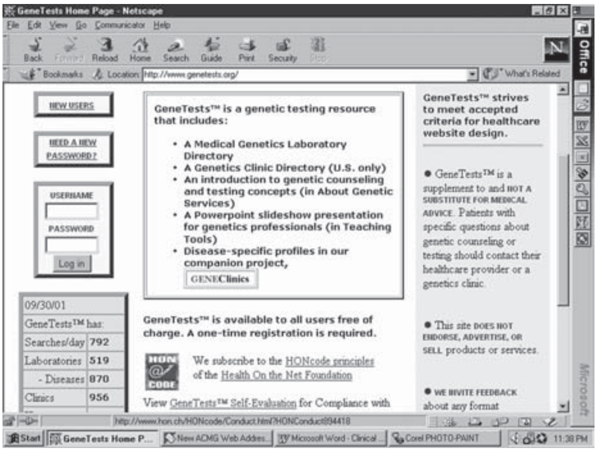
- Diseases (August 2001)
- Features
- Coming Soon

Fig. 5. HGMD Human Gene Mutation Database.

GeneTests

See companion CD for color Figs. 6, 7, and 8

This is a comprehensive and well-curated database of genetic tests and testing laboratories in North America. It is necessary to have a password to enter the database, but this is readily available by following the instructions at the top left of the first page (*see* Fig. 6). Once a user ID and password have been established, entry is obtained by typing these in the boxes at the top left and clicking the **log in**. In the center of the next page, under the blue box, is a link to **Genetics Lab Directory** (*see* Fig. 7). This leads one through a disclaimer page, to a searchable database of genetic laboratories (*see* Fig. 8). Entering **achondro** will return several entries including achondroplasia. Achondroplasia has two boxes next to it **LAB List** and **TEXT Info**. Clicking the **LAB List** box will provide a list of laboratories that provide testing for achondroplasia (including mine in Canada and several in other countries). Clicking the **TEXT Info**


GeneTests Home Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www.genetests.org/ What's Related
Office
Start GeneTests Home P... New ACMG Web Addres... Microsoft Word - Clinical ... Corel PHOTO-PAINT 11:38 PM

NEW USERS

NEED A NEW PASSWORD?

USERNAME

PASSWORD

Log in

09/30/01	
GeneTests™ has:	
Searches/day	792
Laboratories	519
- Diseases	870
Clinics	956

GeneTests™ is a genetic testing resource that includes:

- A Medical Genetics Laboratory Directory
- A Genetics Clinic Directory (U.S. only)
- An introduction to genetic counseling and testing concepts (in About Genetic Services)
- A Powerpoint slideshow presentation for genetics professionals (in Teaching Tools)
- Disease-specific profiles in our companion project, GENEClinics

GeneTests™ is available to all users free of charge. A one-time registration is required.

We subscribe to the HONcode principles of the Health On the Net Foundation

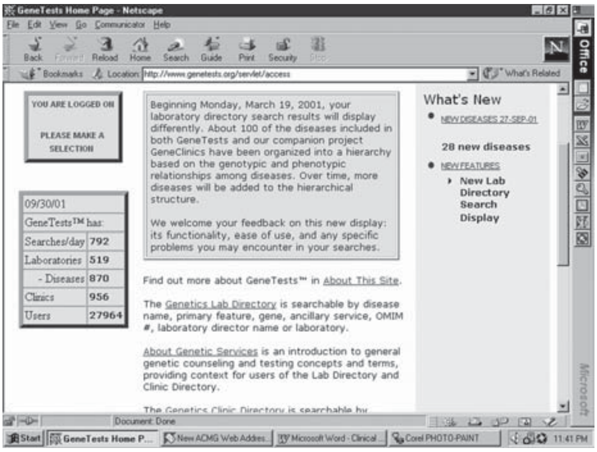
View GeneTests™ Self-Evaluation for Compliance with

GeneTests™ strives to meet accepted criteria for healthcare website design.

- GeneTests™ is a supplement to and NOT A SUBSTITUTE FOR MEDICAL ADVICE. Patients with specific questions about genetic counseling or testing should contact their healthcare provider or a genetics clinic.
- This site DOES NOT ENDORSE, ADVERTISE, OR SELL products or services.
- WE INVITE FEEDBACK about any format

http://www.hon.ch/HONcode/Conduct.html?HONConduct894418

Fig. 6. Oak Ridge National Laboratory (ORNL) Links to the Genetic World.


GeneTests Home Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www.genetests.org/servlet/access What's Related
Office
Document: Done
Start GeneTests Home P... New ACMG Web Addres... Microsoft Word - Clinical ... Corel PHOTO-PAINT 11:41 PM

YOU ARE LOGGED ON

PLEASE MAKE A SELECTION

09/30/01	
GeneTests™ has:	
Searches/day	792
Laboratories	519
- Diseases	870
Clinics	956
Users	27964

Beginning Monday, March 19, 2001, your laboratory directory search results will display differently. About 100 of the diseases included in both GeneTests and our companion project GeneClinics have been organized into a hierarchy based on the genotypic and phenotypic relationships among diseases. Over time, more diseases will be added to the hierarchical structure.

We welcome your feedback on this new display: its functionality, ease of use, and any specific problems you may encounter in your searches.

Find out more about GeneTests™ in About This Site.

The Genetics Lab Directory is searchable by disease name, primary feature, gene, ancillary service, OMIM #, laboratory director name or laboratory.

About Genetic Services is an introduction to general genetic counseling and testing concepts and terms, providing context for users of the Lab Directory and Clinic Directory.

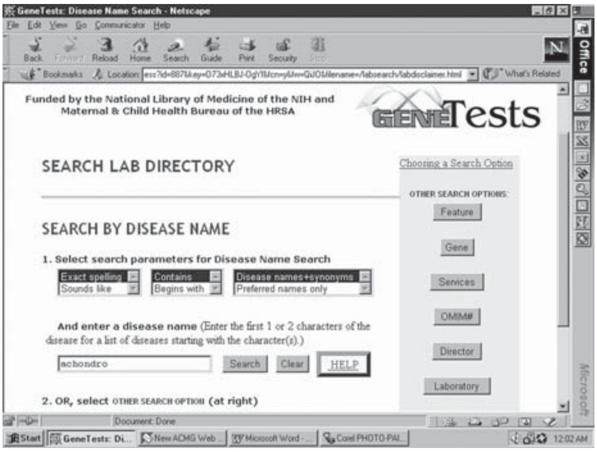
The Genetics Clinic Directory is searchable by

What's New

- NEW DISEASES 27-SEP-01

  28 new diseases
- NEW FEATURES
  - New Lab Directory Search Display

Fig. 7. GeneClinics.


GeneTests: Disease Name Search - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: ess?id=8871&key=073xHLBJ-OgYl&cn=y&lw=QUO&filename=/labsearch/labdisclaimer.html What's Related
Office
GENETests
Document: Done
Start GeneTests: Di... New ACMG Web ... Microsoft Word - ... Corel PHOTO-PAI... 12:02 AM

Funded by the National Library of Medicine of the NIH and Maternal & Child Health Bureau of the HRSA

SEARCH LAB DIRECTORY

SEARCH BY DISEASE NAME

1. Select search parameters for Disease Name Search

Exact spelling / Sounds like; Contains / Begins with; Disease names+synonyms / Preferred names only

And enter a disease name (Enter the first 1 or 2 characters of the disease for a list of diseases starting with the character(s).)

achondro   Search   Clear   HELP

2. OR, select OTHER SEARCH OPTION (at right)

Choosing a Search Option

OTHER SEARCH OPTIONS:

Feature

Gene

Services

OMIM#

Director

Laboratory

Fig. 8. GeneTests Login.

box will take you to the GeneClinics entry for achondroplasia in the database described earlier. On the right-hand side of the search page (*see* Fig. 8), are a series of boxes that give other search options. Clicking the **Director** box on the right and then entering **Bridge** describes this author's laboratory.

EDDNAL

EDDNAL (European Directory of DNA Labs) is a database of genetic tests and testing laboratories in Europe. Clicking the **Start a request** link at the left of the entry page allows one to search the database by entering a McKusick code, i.e., an OMIM number; disease name; contact; e.g., lab director; or country, to retrieve a list of all laboratories in that country (*see* Website http://www.eddnal.com/).

National Colleges

The national college governing the practice of medical genetics in the United States is the American College of Medical Genetics (ACMG). Founded in 1992, the ACMG has grown steadily both in numbers of members and in degree of influence (*see* Website: http://www.acmg.net/). This key website can be accessed for regulations and policy in the United States.

The national college governing the practice of medical genetics in the Canada is the Canadian College of Medical Geneticists (CCMG). Founded in 1975, the CCMG has a long history of leadership in the field of medical genetics. In Canada, the CCMG is also the examining body whereas in the United States this function is separate from the ACMG. (*see* Website: http://ccmg.medical.org/).

Genetic counselors in both the United States and Canada have a specific professional Society or Association. In the United States there is the National Society of Genetic Counselors (NSGC; *see* Website: http://www.nsgc.org/) and in Canada there is the Canadian Association of Genetic Counsellors (CAGC; *see* Website: http://www.cagc-accg.ca/).

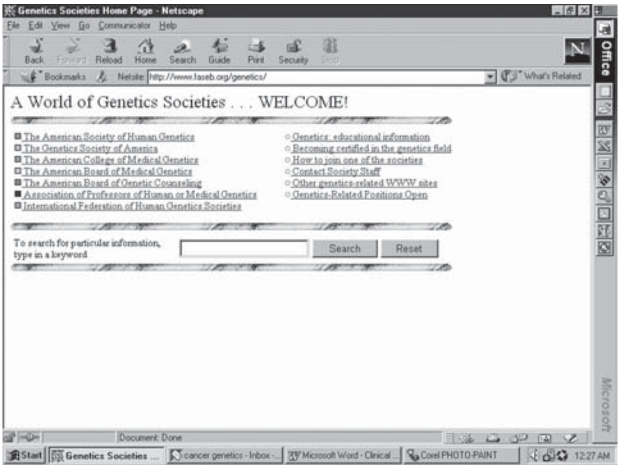
Examining Boards

In the United States, the board examinations are administered by the American Board of Medical Genetics (ABMG; *see* Website: http://www.abmg.org), in Canada, it is administered by the Canadian College of Medical Geneticists (CCMG; *see* Website: http://ccmg.medical. org/), or the Royal College of Physicians and Surgeons of Canada (RCPSC; *see* Website: http://rcpsc.medical.org/). Examinations in genetic counseling are administered by the American Board of Genetic Counseling (ABGC; *see* Website: http://www.abgc.net/) and by the Canadian Association of Genetic Counsellors (CAGC; *see* Website: http://www.cagc-accg.ca/). Information about laboratory technologist certification is available through the Association of Genetic Technologists (AGT; *see* Website: http://www.agt-info.org/) in the United States, the Canadian Society for Medical Laboratory Science (CSMLS; *see* Website: http://www.csmls.org/) in Canada, and the Clinical Molecular Genetics Society (CMGS; *see* Website: http://www.ich.ucl.ac.uk/cmgs/cmgshelp.htm) in the UK. The latter also posts its curriculum on this website.

See companion CD for color Fig. 9

Learned Societies

There is a general starting point for several societies at the Federation of American Societies for Experimental Biology (FASEB; *see* Fig. 9, *see* Website: http://


Genetics Societies Home Page - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Netsite: http://www.faseb.org/genetics/
What's Related
Office
Document: Done
Start Genetics Societies ... cancer genetics - Inbox - ... Microsoft Word - Clinical ... Corel PHOTO-PAINT 12:27 AM
Microsoft

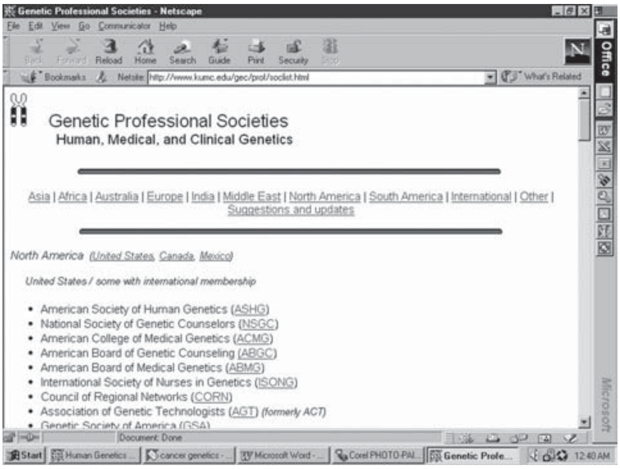
A World of Genetics Societies . . . WELCOME!

- The American Society of Human Genetics
- The Genetics Society of America
- The American College of Medical Genetics
- The American Board of Medical Genetics
- The American Board of Genetic Counseling
- Association of Professors of Human or Medical Genetics
- International Federation of Human Genetics Societies

- Genetics: educational information
- Becoming certified in the genetics field
- How to join one of the societies
- Contact Society Staff
- Other genetics-related WWW sites
- Genetics-Related Positions Open

To search for particular information, type in a keyword

Fig. 9. GeneTests.


Genetic Professional Societies - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Netsite: http://www.kumc.edu/gec/prof/soclist.html
What's Related
Office
Document: Done
Start Human Genetics ... cancer genetics - ... Microsoft Word - ... Corel PHOTO-PAI... Genetic Profe... 12:40 AM
Microsoft

Genetic Professional Societies
Human, Medical, and Clinical Genetics

Asia | Africa | Australia | Europe | India | Middle East | North America | South America | International | Other | Suggestions and updates

North America (*United States, Canada, Mexico*)

United States / some with international membership

- American Society of Human Genetics (ASHG)
- National Society of Genetic Counselors (NSGC)
- American College of Medical Genetics (ACMG)
- American Board of Genetic Counseling (ABGC)
- American Board of Medical Genetics (ABMG)
- International Society of Nurses in Genetics (ISONG)
- Council of Regional Networks (CORN)
- Association of Genetic Technologists (AGT) *(formerly ACT)*
- Genetic Society of America (GSA)

Fig. 10. GeneTests Search for clinical tests provided by various laboratories.

www.faseb.org/genetics/). This organization is still the host of the American Society of Human Genetics (ASHG; *see* Website: http://www.faseb.org/genetics/ashg/ashgmenu.htm) homepage and until very recently also hosted the American College of Medical Genetics (ACMG). A more global perspective can be found at the sites for the European Society of Human Genetics (ESHG; *see* Website: http://www.eshg.org/), the International Federation of Human Genetics Societies (IFHGS; *see* Website: http://www.ifhgs.org/), and the Human Genetics Society of Australasia (HGSA; *see* Website: http://www.hgsa.com.au/). There is a comprehensive listing of societies worldwide to be found at the Genetics Education Center, University of Kansas Medical Center (KUMC; *see* Fig. 10; *see* Website: http://www. kumc.edu/gec/prof/soclist.html).

See companion CD for color Fig. 10

12 Population Genetics

Jill S. Barnholtz-Sloan

Introduction

A species is comprised of many populations of individuals who breed with each other, each with their own unique set of genes (or loci) and alleles. Even so, the population as a whole shares a pool of all genes and alleles. *Evolution* is the change of frequencies of alleles in the total gene pool. Some genotypes, such as those associated with rare and sometimes deadly human diseases, are important to understand from a population genetics perspective so that changes in the incidence of disease can be predicted. Most individuals do not carry the genotypes that cause an extreme *phenotype* (or trait), but of the ones that do carry these rare genotypes or combination of genotypes, their phenotype varies greatly from the average person in the population.

Population genetics is the study of evolutionary genetics at the population level. We study the exchange of alleles and genes within and between populations and the forces that cause or maintain these exchanges. This exchange of genes and alleles causes changes in specific alleles and hence genotype frequencies within and between populations. Through the study of this evolution of alleles and hence, genotype frequencies, we can better understand how to use human populations as a data set to clarify genetic predisposition to disease. A glossary of terms and example calculations is provided at the end of the chapter.

Hardy-Weinberg and Linkage Equilibrium

A basic understanding of the equilibrium that exists in populations is needed in order to begin to understand these changes in allele frequencies over time and across generations. The two types of equilibrium assumed in populations are *Hardy-Weinberg Equilibrium* (HWE) within loci and *Linkage Equilibrium* (LE) between loci.

Fundamental to understanding these equilibria is the understanding that genotype frequencies are determined by mating patterns, with the ideal being random mating. *Random mating* assumes that mating takes place by chance with respect to the locus under consideration. With random mating, the chance that an individual mates with another having a specific genotype is equal to the population frequency of that genotype.

One's *genotype* at any *locus* is made up of two *alleles* (one of two or more forms that can exist at a locus), one allele from the mother and one allele from the father. A

Table 1
Observed and Expected Frequencies of the Genotypes *AA*, *Aa* and *aa* at Hardy-Weinberg Equilibrium After One Mating of Female and Male Gametes, for a Total Sample of *n* Individuals.

		Female Gametes	
		Observed = $p = f(A)$	Observed = $q = f(a)$

Male Gametes	Observed = $p = f(A)$	Observed = $p^2 = f(AA)$
		Expected = np^2
			Observed = $pq = f(Aa)$
			Expected = npq

	Observed = $q = f(a)$	Observed = $pq = f(Aa)$
		Expected = npq
			Observed = $q^2 = f(aa)$
			Expected = nq^2

gene (or *locus*) is the fundamental physical and functional unit of heredity and will carry information from one generation to the next. (Note: Gene and locus will be used interchangeably in this chapter.) By definition, a gene encodes an RNA product that can be structured or encode a protein like an enzyme. The number of alleles that a gene may have can vary from two alleles, a bi-allelic locus, to a large number of alleles, a multi-allelic locus. A genotype can either be homogeneous, i.e., both alleles received from the mother and father are the same allele, or a genotype can be heterogeneous, i.e., the alleles received from the mother and the father are different.

As an example of random mating, consider the locus, *L*, with two alleles *A* and *a*. We would then have three possible genotypes, *AA*, *Aa*, and *aa*. If the population proportions of these genotypes were 0.20, 0.70, and 0.10, respectively, then the chance that a male with *AA* genotype would randomly mate with a female of genotype *AA*, *Aa*, or *aa* is 0.20, 0.70, or 0.10, respectively. These same proportions would apply to female mates with *Aa* or *aa* males. However, in human populations, mating seems to be random with respect to some traits, e.g., blood groups and nonrandom with respect to others, e.g., height, ethnicity, age, and other physical and cultural characteristics.

Independently in 1908, G. H. Hardy and W. Weinberg both published results showing that the frequency of particular genotypes in a sexually reproducing diploid population reaches equilibrium after one generation of random mating and fertilization with no selection, mutation, or migration. The population size is assumed to be very large and the generations are not overlapping. They then showed that the frequencies of the genotypes in the *equilibrium* population are just simple products of the allele frequencies. Again consider locus *L*, has alleles *A* and *a*, with p = frequency $(A) = f(A)$ and $q = f(a) = 1 - p$. Then, if members of the population select their mates at random, without regard to their genotype at the *L* locus, the frequencies of the three genotypes, *AA*, *Aa*, and *aa*, in the population can be expressed in terms of the allele frequencies: $f(AA)=p^2$; $f(Aa) = f(Aa) + f(aA) = 2pq$; $f(aa) = q^2$ (*see* Table 1).

Statistical tests for the presence or absence of HWE for each of the loci of interest are performed using either a *chi-square goodness of fit test* or an *exact test*, to test the null hypothesis that the locus is in HWE versus the alternative hypothesis that the locus is not in HWE. Both tests are testing the fit between the numbers observed and expected numbers, i.e., if the observed is similar enough to the expected, then we accept the null hypothesis of the locus being in HWE. Exact tests are used when there are small sample sizes and/or multi-allelic loci are involved.

Hence, for locus *L*, with alleles *A* and *a*, with frequencies, *p* and *q*, respectively, the observed number of the three possible genotypes are n_{AA}, n_{Aa}, and n_{aa}. Assuming the null hypothesis of HWE is true, we can calculate the expected number for each genotype, *AA*, *Aa*, and *aa*, which are np^2, $2npq$, and nq^2, respectively, where *n* is the total number of individuals in our sample (*see* Table 1). Hence, the chi-square goodness of fit test statistic for locus *L* with one *degree of freedom* (i.e., the total number of possible classes of data = 3, minus the number of parameters estimated from the data = 1, (i.e., the allele frequency *p*), – 1) is

$$\chi_1^2 = \sum_{genotypes} \frac{(Observed - Expected)^2}{Expected} \qquad \text{[Eq. 1]}$$

Exact tests for HWE are more computationally expensive and complicated then the chi-square goodness of fit test. This test is more powerful for multi-allelic loci and small sample sizes than the goodness of fit chi-square test, because it does not depend on large sample asymptotic theory. The exact test for HWE is permutation based and is based on the theory proposed by Zaykin, et al. (1995), and Guo and Thompson (1992). Zaykin, et al. (1995) proposed an algorithm that performs an exact test for disequilibrium between alleles within or between loci. The probability of the set of multi-locus genotypes in a sample, that is conditional on the allelic counts, is calculated from the multinomial theory under the null hypothesis of equilibrium being present. Alleles are then permuted and the conditional probability is calculated for the permuted genotype array. In order to permute the arrays, they employ a Monte-Carlo method. The proportion of permuted arrays that are no more probable than the original sample provides the significance level for the test. Because of the complexity of this testing procedure, a computer program must be used.

Hence, equilibrium refers to the concept that there are no changes in genotypic proportions in a population from generation to generation. This equilibrium will remain constant unless the frequencies of alleles in the population are disrupted. Distorting effects could be any one of the following: selection, migration, nonrandom (assortative) mating and inbreeding, population substructure or subpopulations, mutation or genetic drift.

When we consider a single locus we find two important random-mating principles: 1) genotype frequencies are simple products of the allele frequencies and 2) HWE is reached after one generation of random mating. However, independently, when we consider two or more loci, the first principle is correct for each locus, but not necessarily the second, because the alleles of one locus may not be in random association with the alleles of the second locus. Hence, the state of LE is defined by random allelic association between alleles at any loci. In other words, considering any two loci, the probability that the combination of alleles, one from each locus, is the same if the loci are in the same individual or in different individuals. This state of LE will be reached given enough time, but the approach to equilibrium is slow and highly dependent on the *recombination fraction*, θ.

The distance in centimorgans (cM) between two loci can be estimated from the recombination fraction, θ, between these loci, where θ can be calculated as the probability that the gamete transmitted by an individual is a recombinant, i.e., an individual whose genotype was produced by recombination. After DNA duplication, adjacent chromosomes can change parts, i.e., recombine (*see* Fig. 1), and it is this

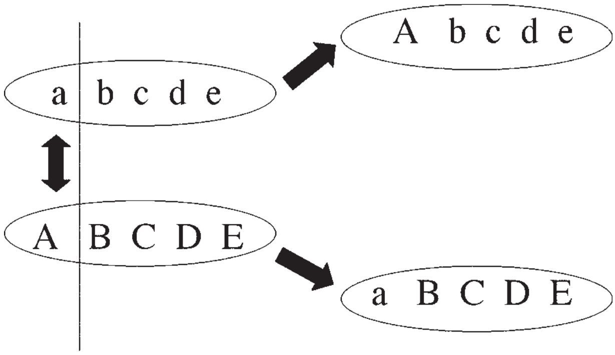


Fig. 1. A pictorial of recombination or crossing over between chromosomal segments (or loci).

genetic shuffling that enables species to have a rich diversity of phenotypic expression from generation to generation in a given population. If the loci are extremely close on the same chromosome, then the likelihood of a crossover between them and the recombination fraction will approach zero. If the loci are far apart or on different chromosomes, then recombination will occur by chance in 50% of meioses and the alleles at one locus will be inherited at random with respect to the alleles at the other locus. Loci with recombination fractions close to zero are tightly linked to each other and loci with recombination fractions of 0.5 are not linked to each other.

Thus, allelic *linkage disequilibrium* (*LD*) is measured by a statistic D, which is defined as $D = P_{ij} – P_i q_j$, where P_{ij} is the frequency of the gamete carrying i^{th} allele of one locus and the j^{th} allele of another and p_i and q_j are the frequencies of the i^{th} and j^{th} alleles of the two loci. D can be a positive or negative number or zero. In random mating populations with no selection, LD is reduced in every generation at a rate of θ (recombination fraction), $0 < \theta < 0.5$; leading to $D(t) = (1 – \theta)^t d(0)$, where $D(t)$ is the disequilibrium at generation t and $D(0)$ is the disequilibrium at generation zero.

Tests for significant allelic LD for combinations of alleles from two and three loci (different combinations of the alleles of the loci of interest that are adjacent to each other on a chromosome), are performed using a chi-square test to test the null hypothesis that $d(ij) = 0$ versus the alternative that $d(ij) = 0$. In order to calculate an allelic disequilibrium measure, $d(.)$, allele and genotype/haplotype (the combination of several alleles from multiple loci), frequencies must first be calculated. This can be done very simply by a direct counting method of the possible genotypes/haplotypes present in the population of interest, which can then be used to assess allele frequencies.

Let us assume that locus 1 has i alleles, locus 2 has j alleles, and locus 3 has k alleles and that the loci lie adjacent to each other. Then the calculations for combinations of alleles (haplotypes) at the three loci are as follows:

$$d(ij) = f(ij) – f(i)f(j) = f(jk) – f(j)f(k)$$
$$d(ijk) = f(ijk) – f(i)d(ij) – f(j)d(ik) – f(i)f(j)f(k)$$

where

$d(ij)$ = disequilibrium of allele i at locus 1, and allele j at locus 2;
$d(jk)$ = disequilibrium of allele j at locus 2, and allele k at locus 3;
$d(ijk)$ = disequilibrium of allele i at locus 1, and allele j at locus 2, allele k at locus 3;
$f(i)$ = frequency of allele i at locus 1, $f(j)$ = frequency of allele j at locus 2;

$f(k)$ = frequency of allele k at locus 3;
$f(ij)$ = frequency of haplotype ij (locus 1 and locus 2);
$f(jk)$ = frequency of haplotype jk (locus 2 and locus 3);
$f(ijk)$ = frequency of haplotype ijk (considering all three loci);

which would yield calculations for d (allele i at locus 1 and allele j at locus 2), d (allele j at locus 2 and allele k at locus 3) and d (alleles i, j, and k at loci 1, 2, and 3). The significance of the two locus disequilibrium is tested using an allele specific chi-square test, which tests whether or not d (allele i at locus 1 and allele j at locus 2) and d (allele j at locus 2 and allele k at locus 3) are significantly different from zero. The two-locus test used is the test proposed by Hill (1974), given by

$$\chi_{ij}^2 = \frac{n[d(ij)]^2}{f(i)\,[1-f(i)]\,f(j)\,[1-f(j)]} \qquad \text{[Eq. 2]}$$

which follows a χ^2 distribution with one degree of freedom.

For the three locus test, the test statistic used is the one suggested by Weir (1996), given by

$$\chi_{ijk}^2 = \frac{[d(ijk)]^2}{Var\,[d(ijk)]}\text{, with one degree of freedom}$$

where

$$Var\,[d(ijk)] = \frac{1}{n}\left[\begin{array}{l}\pi_i\pi_j\pi_k + 6d(ij)d(jk)d(ik) + \pi_i\left(\tau_j\tau_k d(jk) - [d(jk)]^2\right) + \pi_j\left(\tau_i\tau_k d(ik) - [d(ik)]^2\right) \\ + \pi_k\left(\tau_i\tau_j d(ij) - [d(ij)]^2\right) + d(ijk)\left[\tau_i\tau_j\tau_k - 2\tau_i d(jk) - 2\tau_j d(ik) - 2\tau_k d(ij) - d(ijk)\right]\end{array}\right]$$

in which

$$\begin{array}{l}\pi_i = f(i)[1-f(i)];\ \pi_j = f(j)[1-f(j)];\ \pi_k = f(k)[1-f(k)] \\ \tau_i = [1-2f(i)];\ \tau_j = [1-2f(j)];\ \tau_k = [1-2f(k)]\end{array} \qquad \text{[Eq. 3]}$$

which tests whether d (alleles i, j, and k at loci 1, 2, and 3) is significantly different from zero and also follows a chi-square distribution.

Linkage disequilibrium can be the result of many different circumstances. LD could have occurred in the founding population and because of a very small θ, has not had sufficient time (in generations of random mating) to disappear. The loci could be tightly linked, so that recombinants are rare, causing the approach to equilibrium to be slow. *Population admixture*, or the matings of different subpopulations with different allele frequencies, can also cause LD. Interaction between the loci of interest can cause LD to be present because of the proximity of the loci, i.e., the loci are closely linked. Selection of specific heterozygotes can also overcome the natural tendency for D to go to zero. Lastly, LD can be caused purely by chance, in that some loci may present themselves in a higher frequency in a population and stay at that frequency. For further information on estimating and testing of LD refer to Hill (1974) and Weir (1996).

Darwin and Natural Selection

The previously mentioned assumptions of equilibrium and hence allele and genotype frequencies, are all directly affected by the forces of evolution that exist all around us, such as natural selection, mutation, genetic drift and mutation, inbreeding, nonrandom mating, and population structure. The mechanism of evolution was the subject of speculation by several individuals in the early nineteenth century, but it was Charles Darwin who took up this problem. He proposed that the cause of evolution was natural selection in the presence of variation; *natural selection* is the process by which the environment limits population size.

He based this theory on three key observations: 1) When conditions allow individuals in a population to survive and reproduce, they will have more offspring than can possibly survive (population size increases exponentially); 2) Individuals will vary in their ability to survive and reproduce, most of the time because of a lack of available resources; and 3) No two individuals are the same because of variation in inherited characteristics, therefore they all vary in their ability to reproduce and survive. From these observations he deduced the following: 1) There is competition for survival and successful reproduction; 2) Heritable differences that are favorable will allow those individuals to survive and reproduce more efficiently as compared to individuals with unfavorable characteristics; i.e., elimination is selective; and 3) Subsequent generations of a population will have a higher proportion of favorable alleles present than previous generations. With the increase of these favorable alleles in the population, comes an increase of the favorable genotype(s), so that the population gradually changes and becomes better adapted to the environment. This is the definition of *fitness*; genotypes with greater fitness produce more offspring than less fit genotypes. Fitness of a gene or allele is directly related to its ability to be transmitted from one generation to the next.

Individuals are forced to compete for resources in order to stay alive and reproduce successfully, therefore, certain genotypes that are the genetic determinants for the more favorable characteristics in a population, will become more common than the less favorable genotypes. As a result, different genotypes will have different likelihoods of success. The effects of relative frequencies of the genotypes, group interactions and environmental effects can complicate this likelihood. Sexual selection is a key component to the likelihood of success and can be affected by direct competition between individuals of the same sex for a mate of the opposite sex or by mating success, which is a direct result of the choice of a mate.

There are three modes of natural selection (*see* Fig. 2): 1) *Stabilizing selection*: this removes individuals who deviate too far from the average and maintains an optimal population, i.e., selection for the average individual; 2) *Directional selection*: this favors individuals who are at one extreme of the population, i.e., selection of either of the extreme individuals; and 3) *Disruptive selection*: this favors individuals at both extremes of the population, which can cause the population to break into two separate populations.

Neo-Darwinism arose when it was realized that variation is a direct result of spontaneous mutation. The theoretical basis of Neo-Darwinism was then developed, based on a mathematical framework and has become essential to understanding molecular evolution. From the 1930s to the 1950s researchers worked to try to better understand the empirical basis of Neo-Darwinism, but this was met with great difficulty, because a human's lifetime is generally not long enough to be able to observe substantial changes in populations.

Types of Variation

Darwin's work and the work of the Neo-Darwinists helped us better understand that the variation within and between populations is caused and maintained by mutation, genetic drift, migration, inbreeding, and nonrandom mating, including the types of natural selection discussed in the previous section. A summary of whether or not each of these components of evolution increases or decreases variation within and between populations is given in Table 2.

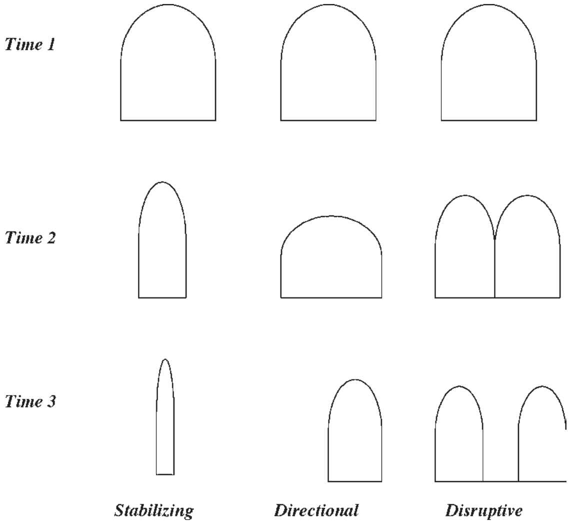


Fig. 2. Three types of natural selection—stabilizing, directional and disruptive—over the course of three different time periods 1, 2, and 3 (three subsequent generations of mating) and their effects on the normally distributed initial population in time period 1.

Table 2
The Effect of the Different Forces[a] of Evolution on Variation Within and Between Populations

Evolutionary component	*Within populations*	*Between populations*
Inbreeding and nonrandom mating	Decrease	Increase
Genetic drift	Decrease	Increase
Mutation	Increase	Decrease
Migration	Increase	Decrease
Selection		
– Stabilizing	Increase	Decrease
– Directional	Decrease	Increase and decrease
– Disruptive	Decrease	Increase

[a] Either an increase or decrease of variation within and/or between populations is shown for each force.

In 1953, the Watson-Crick model of DNA (deoxyribonucleic acid) was put forward, opening doors to the application of various molecular techniques in population genetics research. Because DNA is the chemical substance that encodes all genes, relationships between and within populations could now be characterized through the study of DNA. Now researchers could study the variation within a species instead of having to study the species as a whole. Researchers began by studying amino acid changes. With the advent of electrophoresis, a simpler method of studying protein variation, they then switched to studying genetic polymorphism within populations in the mid 1960s, Many other technical breakthroughs have since emerged such as restriction enzyme digestion, gene cloning, and rapid DNA sequencing. Together, these tools have uncovered many unexpected properties of the structure and organization of genes.

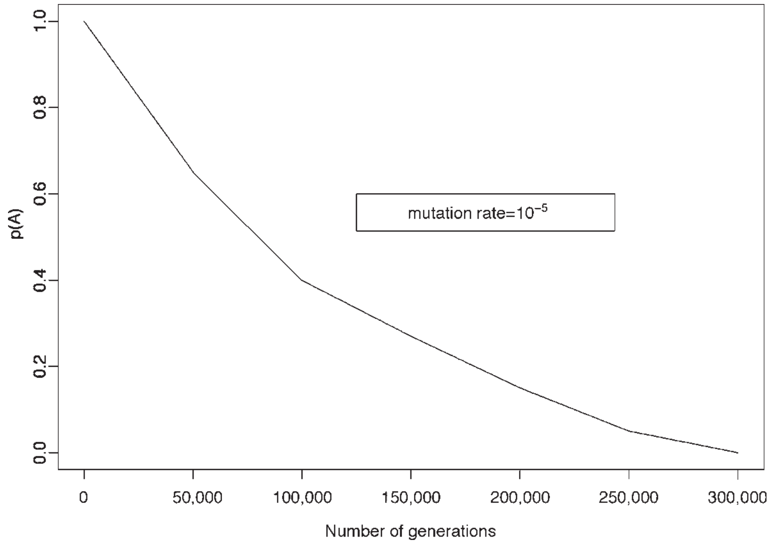


Fig. 3. The cumulative effect of mutation over generations of mating (over time) on the change in frequency of allele *A*, where the mutation rate for *A* to become *a*, is maintained at a constant rate of 10^{-5}.

Mutation

Through these new methods of molecular study, it was discovered that all new variation begins with a *mutation* or a change in the sequence of the bases in the DNA. A mutation in the DNA sequence caused by nucleotide substitution (e.g., sickle cell hemoglobin production), insertions/deletions (e.g., hemophilia A and B, cystic fibrosis, *cru-du-chat* syndrome), triplet expansion (e.g., Huntington's disease), translocation (e.g., Down's syndrome), etc., may be spread through the population by genetic drift and/or natural selection and eventually become fixed in a species. If this mutant gene produces a new phenotype, this new characteristic or trait will be inherited by all subsequent generations unless the gene mutates again. Some mutations will not affect the protein product and these are called silent mutations. Some mutations will occur in noncoding regions, which may or may not have regulatory roles. Hence, it is the regions where the sequence is important for function where variation is the most interesting in terms of effects on the population.

Spontaneous mutation rates are appreciably small, 10^{-4}–10^{-6} mutations per gene per generation, so it is the cumulative effects of mutation over long periods of time that become important (*see* Fig. 3). The simplest kind of mutation is when one nucleotide is replaced by another (a base substitution). These substitutions can be transitions, A to G or C to T, or transversions, all other types of substitutions. If a base substitution results in the replacement of an amino acid in the protein product, this is a missense mutation. Mutations can also result in a loss or gain of genetic material, deletion or insertion, which can result in frame-shift mutations. Genetic material can also be rearranged, by translocation, where pieces of different chromosomes change places with one another. Mutations owing to gene conversion come from the misalignment of DNA, which is associated with the unequal crossing over of parts of adjacent chromosomes.

Table 3
Genotypic Frequencies of the Three Genotypes Present in the *MN* Blood Group; *MM*, *MN* and *NN*, Calculated from a Total of 1419 Individuals

Genotype	*Number of Individuals*	*Genotypic Frequencies*
MM	543	543/1419 = 0.38
MN	419	419/1419 = 0.30
NN	457	457/1419 = 0.32
Total	1419	1.0

With the purpose of better grasping the concept of mutation, tabulating correct allele and genotype frequencies for a population under study is imperative. A basic understanding of polymorphisms, heterozygosity, and gene diversity is also critical.

Allele and Genotype Frequencies

The most fundamental quantitative variable in the study of population genetics is the allele frequency. In a population of N diploid individuals we have $2N$ alleles present. If the number of alleles i in the population is n_i, then the frequency of that allele in the population is defined as $p_i = n_i / 2N$. There is no limitation to the number of alleles that may exist at a single locus but their frequencies must always sum to one. When a locus has only two alleles we denote their frequencies as p and $q = 1 - p$. A biallelic locus, L, with alleles A and a, has three possible genotypes, *AA*, *Aa*, and *aa*. However, not all genotypes are necessarily present at all times in a population.

The genotype frequencies at a particular locus are defined similarly; the frequency of a particular genotype is the number of that genotype present in the population divided by the total number of genotypes present. Like the allele frequencies, genotype frequencies must sum to one over all genotypes present in the study population. However, the number of genotypes is constrained and equals $[m(m + 1)]/2$ if there are m alleles at the locus: m homozygotes and $[m(m - 1)]/2$ heterozygotes.

As an example of how to count alleles and genotypes in a population, the *MN* blood group will be used. Assume that a population consists of 543 *MM* (*phenotype M*), 419 *MN* (*phenotype MN*), and 457 *NN* (*phenotype NN*) individuals (total = 1419 individuals). We then need to determine the values of $p = f(M)$ and $q = f(N)$ and the genotype frequencies in the population. In this simple example, the values of p and q and the genotype frequencies can be determined by counting the genotypes and alleles. To determine the genotypic frequencies we simply divide the number of each genotype present in the population by the total number of individuals present in the population (*see* Table 3), where the genotypic frequencies will add to one. To determine the allelic frequencies we simply count the number of M or N alleles and divide by the total number of alleles. There are 543 *MM* individuals, which means there are 543 × 2 = 1086 M alleles being contributed. 419 *MN* individuals will contribute 419 M alleles and 419 N alleles. And there are 457 *NN* individuals contributing 457 × 2 = 914 N alleles. Therefore, there are 1086 + 419 = 1505 M alleles total in the population and 419 + 914 = 1333 N alleles total in the population. So, $p = f(M) = 1505/[2(1419)] = 0.53$ and $q = f(N) = 1333/[2(1419)] = 0.47 = 1 - p$.

Polymorphism

When a locus has many variants, or alleles, it is referred to as being *polymorphic*. Polymorphism is defined as the existence of two or more alleles with large relative frequencies in a population (occurrence of no less than 1–2%). The limiting frequency of the most common allele, and thus for the polymorphism, is set at 99%. The existence of these multiple alleles is generated by mutation(s) and recombination at a locus. Most are eliminated from the population by genetic drift or purifying selection and only a small number of them are incorporated into the population by chance or selection. The first human polymorphism discovered was the ABO blood group by Landsteiner (1900). Most polymorphisms are genetically simple, in that two alleles directly determine two versions of the same protein. Others can be highly complex, with multiple, related loci engaged in a complex system on a chromosome.

There are four primary ways to determine polymorphisms: Restriction Fragment Length Polymorphisms (RFLPs), *Minisatellites* or Variable Number of Tandem Repeats (VNTRs), *Microsatellites* or Short Tandem Repeats (STRs) or Single Nucleotide Polymorphisms (SNPs). RFLPs are DNA segments of different lengths generated by restriction enzyme digestion, targeted to specific base sequences. The different-sized DNA fragments can be separated using electrophoresis. Because RFLPs are based on single nucleotide changes, they are not very polymorphic in the population and usually have heterozygosities of less than 50%. Minisatellites or VNTRS are repeats of a relatively short oligonucleotide sequence that vary in number from one person to another. They are much more polymorphic than RFLPs. Micro-satellites or STRS are multiple (often 100 or more) repeats of very short sequences (2–4 nucleotides), e.g., $(CA)_n$ repeats that are amplified by polymerase chain reaction (PCR) and electrophoresed to score allele sizes. These are highly polymorphic in the population, with most individuals being heterozygous. Thousands of such markers are available, and are conveniently located throughout the genome. Tri- and tetranucleotide repeats are often easier to score than dinucleotide repeats. Microsatellites are often the markers of choice for genetic studies. SNPs are a class of recently identified markers characterized by variation at a specific nucleotide. Only 2 alleles exist for a given SNP in the population, therefore they are not as polymorphic as microsatellites. However, they are abundant throughout the genome.

Gene Diversity and Heterozygosity

When a large number of loci are examined in a population for variation, the amount of variation is usually measured by the proportion of polymorphic loci. This can be reported for a single locus, as an average over several loci or as the *average heterozygosity* per locus or *gene diversity*. The *heterozygosity* (the proportion of heterozygotes or polymorphic loci) is defined purely in terms of genotype frequencies in the population. If n_{ij} is the observed count of heterozygotes *ij* at locus *L*, where *i* and *j* are different alleles, in a sample of size *n*, then the sample heterozygosity for that locus *L* is given by

$$H_L = \sum_i \sum_{i \neq j} \frac{n_{ij}}{n} \qquad \text{[Eq. 4]}$$

Heterozygosity is calculated separately for each locus under study and then averaged over all loci under consideration (*m*), to give

$$\bar{H} = \frac{1}{m} \sum_{l=1}^{m} H_L \qquad \text{[Eq. 5]}$$

Average heterozygosity or gene diversity is a more useful measure of variation than the proportion of heterozygotes (heterozygosity) because it is not subject to bias caused by sample size, whether it be the size of the study sample or the number of loci being examined. Also, it is calculated from allele frequencies and not genotype frequencies. Assume that p_j is the frequency of the j^{th} allele at the l^{th} locus, then the gene diversity at this locus, L, is

$$D_L = 1 - \sum_j p_j^2 \qquad \text{[Eq. 6]}$$

and as an average over m loci ,

$$\bar{D} = 1 - \frac{1}{m} \sum_l \sum_j p_{jl}^2 \qquad \text{[Eq. 7]}$$

where $l = 1,...,m$. In a randomly mating population, $\bar{D}$ is equal to the average proportion of heterozygotes per locus. Hence, a very polymorphic locus will have a higher gene diversity, because, as more alleles are present at a locus, more heterozygotes will be possible.

However, it is not just mutation that is responsible for sustaining variation. Natural selection, genetic drift, and migration also play key roles in its maintenance, as well as inbreeding and nonrandom mating and the genetic structure of a population. Selection acts against the dysfunctional alleles that are continuously created by mutation. At equilibrium, the number of new dysfunctional mutations equals the number lost by selection. Selection, in fact, favors heterozygotes, because they maintain two different alleles in their genotypes and rare alleles are more common in a heterozygous individual. Hence, the heterozygous individual can carry more information than the homozygous individual.

Genetic Drift and Migration

Genetic drift is the change in allele frequency that results from the chance difference in transmission of alleles between generations. The gene pool changes at each generation, so that the frequencies of particular alleles will change (drift) through time, and these frequencies can go up or down, accumulating with time. Drift's largest effects are seen on small populations (larger samples will be closer to the average) and on rare alleles (the transmission frequency will be higher than the expected average frequency if it is a common allele). Drift is important because it has a greater effect on transmission of rare alleles than selection, as it helps remove or promote very rare alleles.

In small populations, drift can cause certain allele frequencies to be much larger or smaller than would likely occur in a large population. This is called the *founder effect*, when a small, underrepresented group forms a new colony. The Amish in the United States are a good example because the roots of this population can be traced to a small number of immigrant families. When a population is reduced to a small number, possibly because of disease, and later becomes the parents of a new large population, this is called *bottle-necking*. Drift can also cause small isolated populations to be very different from the norm, which can lead to the formation of new species and races.

The basic calibrator of genetic drift is *effective population size*, N_e. This is the size of a homogeneous population of breeding individuals, half of which are male and half are female, that would generate the same rate of allelic fixation as observed in the real population of total size N. Hence, in the real population of size N, the variance of the random deviation of allele frequencies is $[p(1 - p)]/2N$ and the rate of decay is $1/2N$. A human population is structured in many different ways, i.e., its individuals are of different sexes, ages, and geographical and social groups. This does not match the "ideal" population. Accordingly, N_e is estimated indirectly to be $N/2$ to $N/3$, but can be estimated directly if the heterozygosity or the inbreeding coefficient of the population is known.

Migration also causes variation within populations, because of the possibility of mixing many populations together. Geographically defined populations generally show variation between each other and such variation can effect the fate of that population. Through migration, these populations subdivide and mix with new individuals to form new, sustainable populations. Population subdivision causes a decrease in homozygous individuals, known as *Wahlund's principle*, because it will increase variation in this newly formed population. In human populations, the main effect of this fusion of populations is that it will decrease the overall frequency of children born with genetic defects resulting from homozygous recessive genes that have a high frequency in one of the mixing populations.

Wright's Fixation Indices

Genetic structure of a species is characterized by the number of populations within it, the frequencies of the different alleles in each population and the degree of genetic isolation of the populations. The evolutionary forces previously discussed will cause differentiation within and between subpopulations within a larger species population. Wright (1931, 1943, 1951) showed that any species population has three levels of complexity, I, the individual; S, the various subpopulations within the total population and T, the total population. In order to assess this population substructure and test for allelic correlation within subpopulations, Wright defined three measurements called *fixation indices* that have correlational interpretations for genetic structure and are a function of heterozygosity. F_{IT} is the correlation of alleles within individuals over all subpopulations; F_{ST} is the correlation of alleles of different individuals in the same subpopulation; and F_{IS} is the correlation of alleles within individuals within one subpopulation. Cockerham (1969, 1973) later showed analogous measures for these three fixation indices, which he called the overall inbreeding coefficient, F (i.e., F_{IT}), the coancestry, θ (i.e., F_{ST}) and, the inbreeding coefficient within subpopulations, f (i.e., F_{IS}). These are all related by the following formula:

$$f = \frac{F - \theta}{1 - \theta} \qquad \text{[Eq. 8]}$$

In order to calculate the three fixation indices, we must first calculate the heterozygosities. H_I is the heterozygosity of an individual in a subpopulation and can be interpreted as the average heterozygosity of all the genes in an individual. H_S is the expected heterozygosity of an individual in another subpopulation and can be interpreted as the amount of heterozygosity in any subpopulation if it were undergoing random mating. H_T is the expected heterozygosity of an individual in an equivalent random mating total population and can be interpreted as the amount of heterozygosity in a total population where all subpopulations were pooled together and mated randomly.

Then if H_i is the heterozygosity in subpopulation i, and if we have k subpopulations,

$$H_I = \sum_{i=1}^{k} \frac{H_i}{k} \qquad \text{[Eq. 9]}$$

If p_{js} is the frequency of the j^{th} allele in subpopulation s, then H_S is the expected Hardy-Weinberg heterozygosity in subpopulation s,

$$H_S = 1 - \sum_{i=1}^{h} p_{js}^2 , \qquad \text{[Eq. 10]}$$

for a total of h alleles at that locus. $\bar{H}_s$ is the average taken over all subpopulations. Finally if $\bar{p}_j$ is the frequency of the j^{th} allele, averaged over all subpopulations,

$$H_T = 1 - \sum_{i=1}^{k} \bar{p}_j^2 , \qquad \text{[Eq. 11]}$$

for all k subpopulations. Thus, Wright's F statistics are

$$F_{IS} = \frac{\bar{H}_S - H_I}{\bar{H}_S} ; F_{ST} = \frac{H_T - \bar{H}_S}{H_T} ; F_{IT} = \frac{H_T - H_I}{H_T} \qquad \text{[Eq. 12]}$$

and these three equations are related by the following identity,

$$(1 - F_{IS})(1 - F_{ST}) = (1 - F_{IT}) \qquad \text{[Eq. 13]}$$

Genetic Distance

The degree of genetic isolation of one subpopulation from another can be measured by *genetic distance*, which can be interpreted as the time since the subpopulations that are under comparison diverged from their original ancestral population. Nei proposed the most widely used measure of genetic distance between subpopulations in 1972, even though the concept of genetic distance was first used by Sanghvi (1953). The mathematics of these measurements was later refined by Fisher (1963) and Mahalanobis (1963). Thus, Nei's standard genetic distance is given by $D = -ln(I)$, where I is called the genetic identity, which corrected for bias, is calculated using the following equation:

$$I = \frac{(2n-1)\sum_l \sum_j p_{j1} p_{j2}}{\sqrt{\sum_l \left(2n \sum_j p_{j1}^2 - 1\right) \sum_l \left(2n \sum_j p_{j1}^2 - 1\right)}} \qquad \text{[Eq. 14]}$$

where we are examining the j^{th} allele at the l^{th} locus for populations 1 and 2 and p_{j1} is the frequency of the j^{th} allele at the l^{th} locus for population 1 and p_{j2} is the frequency of the j^{th} allele at the l^{th} locus for population 2, from a total sample of n individuals.

Inbreeding and Non-Random Mating (Assortative Mating)

Inbreeding and other forms of *nonrandom mating*, or *assortative mating*, can also have a profound effect on variation within a population. Inbreeding refers to mating between related individuals. Inbreeding is when genetically similar (related) individuals mate more frequently than would be expected in a randomly mating population. Inbreeding mainly causes departures from HWE and as a consequence of this departure from equilibrium, an increase in homozygotes. Inbreeding can cause the offspring of the mating to have replicates of specific alleles present in the shared ancestor of the parents. Thus, inbred individuals may carry two copies of an allele at a locus that are *identical by descent* (IBD) from a common ancestor. The proportion of IBD is how frequently two offspring share copies of the same parental (ancestral) allele.

The amount of inbreeding in a population can be measured by comparing the actual proportion of heterozygous genotypes in the population that is inbreeding to the pro-

Table 4
Frequencies of Genotypes *AA*, *Aa*, and *aa* After One Generation of Inbreeding Where the Reference Population is the Preceding Generation Before the Inbreeding[a]

	Origin			
Genotype	*Independent*	*Identical by descent*	*Original frequencies*	*Frequency change after inbreeding*
AA	$p^2(1 - F)$	$+ pF$	$= p^2$	$+ pqF$
Aa	$2pq(1 - F)$		$= 2pq$	$- 2pqF$
Aa	$q^2(1 - F)$	$+ qF$	$= q^2$	$+ pqF$

[a] The inbreeding coefficient, F and the two different types of origin, independent and identical by descent, are incorporated in the calculations.

portion in a randomly mating population. Inbreeding alone cannot change allele frequencies, but it can change how the alleles come together to form genotypes.

In order to illustrate this change in genotype frequencies, a simple case of inbreeding will be examined, where the reference population will be the preceding generation so that the inbreeding coefficient, F, measures the increase in IBD from one generation to the next. If allele A has a frequency of p and allele a has a frequency of $q = 1 - p$, then the frequency of AA genotypes in an inbred gamete will be $f(AA) = p^2(1 - F) + pF = p^2 + pqF$. An individual of genotype AA can be formed by one of two ways, of independent origin, which has a probability of $p(1 - F)$ and identical by descent, which has a probability of F. Therefore, if $F > 0$, there will be an excess of AA homozygotes relative to what would be expected by HWE. If $F = 0$, then the frequency of the AA homozygotes will be as is expected by HWE. Similarly, the frequency of the homozygote aa would be $f(aa) = q^2(1 - F) + qF = q^2 + pqF$. And the same rules would hold for the relationship between the value of F and HWE. The probability of the heterozygote Aa is more complicated to calculate. But, we know that the frequencies of the genotypes must sum to one so, $f(Aa) = 1 - f(AA) - f(aa) = 2pq(1 - F) = 2pq - 2pqF$. Table 4 shows a summary of the changes in genotype frequencies after one generation of inbreeding.

If natural selection and inbreeding act together they can have a profound effect on the course of evolution because of the increase in the frequency of homozygous genotypes. Inbreeding in human populations can result in a much higher frequency of recessive disease homozygotes, since recessive disease alleles generally have low frequencies in humans. Inbreeding affects all alleles and hence, genes, in inbred individuals. This exposes rare recessive disorders that may not have presented if inbreeding did not occur.

Nonrandom mating or assortative mating is when a mate is chosen based on a certain phenotype. In other words, it is the situation when mates are more similar (or dissimilar) to each other than would be expected by chance in a randomly mating population. In positive assortative mating, a mate chooses a mate that phenotypically resembles himself or herself. In negative assortative mating, a mate chooses a mate that is phenotypically very different from himself or herself. Assortative mating will only affect the alleles that are responsible for the phenotypes affecting mating frequencies. The genetic variance (or variability) of the trait that is associated with the mating increases with more generations of assortative mating for that trait. In humans, positive assortative mating occurs for traits like intelligence (IQ score), height, or certain socioeconomic variables. Negative assortative mating occurs mostly in plants.

Conclusions

Even with the advances in molecular biology and genetics, all of this discovery in the field of population genetics and of the characteristics that cause population-based changes and their consequences, i.e., how they can affect human disease susceptibility, had until recently only been assessed one at a time by a technician in a laboratory. Now with the advent of the gene chips or microarrays, these new methods can be automated and carried out at a much larger scale, e.g., 10,000 genotypes can be determined using a single gene chip. These and other faster techniques will permit all genes to be tested for polymorphism within and between populations for many individuals in a population at a time and many populations at a time. This new technology will allow us even greater insight into the relationship between human genetic changes in populations and evolutionary forces.

Glossary of Terms

Allele One of two or more forms of a gene or locus that can exist at a locus (variants of a locus).

Average Heterozygosity (or Gene Diversity) The average proportion of heterozygotes in the population and the expected proportion of heterozygous loci in a randomly chosen individual.

Bottle-Necking When a population is reduced to a small number, possibly because of disease, and later becomes the parents of a new large population.

Chi-Square Goodness of Fit Test A statistical test used to test the fit between the observed and expected numbers in a population; the test statistic has a chi-square distribution.

Degrees of Freedom The number of possible classes of data, minus the numbers of parameters estimated from the data, minus 1.

Directional Selection Favors individuals who are at one extreme of the population, i.e., selection of either of the extreme individuals.

Disruptive Selection Favors individuals at both extremes of the population, which can cause the population to break into two separate populations.

Effective Population Size, N_e The size of a homogeneous population of breeding individuals, half male and half female, that would generate the same rate of allelic fixation as observed in real population of total size N.

Evolution The change of frequencies of alleles in the total gene pool.

Exact Test A statistical test used when the sample sizes are small or the locus under study is multi-allelic. It is more powerful than the chi-square goodness of fit test.

Fitness The ability of a gene or locus to be transmitted from one generation to the next; genotypes with greater fitness produce more offspring than less fit genotypes.

Founder Effect when a small, underrepresented group forms a new colony.

Gene (or locus) The fundamental physical and functional unit of heredity that will carry information from one generation to the next; generally encodes for some gene product, like an enzyme or protein. It is comprised of at least two alleles, a bi-allelic locus, to a large number of alleles, a multi-allelic locus. (Note: gene and locus are used interchangeably).

Genetic Distance The degree of genetic isolation of one subpopulation from another; interpreted as the time since the subpopulations that are under comparison diverged from their original ancestral population.

Genetic Drift The change in allele frequency that results from the chance difference in transmission of alleles between generations.

Genotype Composed of two alleles at a particular locus, one from the mother and one from the father. Homogeneous genotype, i.e., both alleles received from the mother and father are the same allele or heterogeneous genotype, i.e., the alleles received from the mother and the father are different.

Haplotype The combination of several alleles from multiple loci.

Hardy-Weinberg Equilibrium (HWE) The frequencies of the genotypes in the equilibrium population. Simple products of the allele frequencies.

Heterozygosity The proportion of heterozygotes or polymorphic loci in a population.

Identical by Descent (IBD) How frequently two offspring share copies of the same parental (ancestral) allele.

Inbreeding When genetically similar (related) individuals mate more frequently than would be expected in a randomly mating population.

Linkage Equilibrium (LE) Random allelic association between alleles at any loci. Considering any two loci. The probability of the combination of alleles, one from each locus, is the same if the loci are in the same individual or in different individuals.

Migration The movement of individuals within and between populations.

Mutation A change in the sequence of the bases in the DNA.

Natural Selection The process by which the environment limits population size.

Non-random Mating (or Assortative Mating) Mating does not take place at random with respect to some traits. Mates can chose each other based on height, ethnicity, age, and other physical and cultural characteristics.

Phenotype Trait or characteristic of interest.

Polymorphism The existence of two or more alleles with large relative frequencies in a population (occurrence of no less than 1–2%); limiting frequency is 99%; four ways to determine: Restriction Fragment Length Polymorphisms (RFLPs); *Minisatellites* or Variable Number of Tandem Repeats (VNTRs); *Microsatellites*, or Short Tandem Repeats (STRs), or Single Nucleotide Polymorphisms (SNPs).

Population Admixture The matings of different subpopulations with different allele frequencies.

Population Genetics The study of evolutionary genetics at the population level.

Random Mating Mating takes place by chance with respect to the locus under consideration. The chance that an individual mates with another having a specific genotype is equal to the population frequency of that genotype.

Recombination Fraction (θ) The probability that the gamete transmitted by an individual is a recombinant, i.e., an individual whose genotype was produced by recombination.

Stabilizing Selection Removes individuals who deviate too far from the average. It maintains an optimal population, i.e., selection for the average individual.

Wahlund's Principle A decrease in homozygous individuals in a population, caused by population subdivision.

Example Calculations

Disease Allele Frequency Calculations Based on Hardy-Weinberg Proportions

Example 1:
Tay-Sachs Disease (autosomal recessive; q is disease allele):

a. U.S. Ashkenazi incidence: 1/3,900 = q^2
$q = 0.016$, $p = 1 - 0.016 = 0.984$
$p^2 = 0.968$, $2pq = 0.031$, $q^2 = 0.00026$
(Sum = 0.99926 ~ 1.0)

Gene frequency = $q = 0.016$,
Carrier frequency = $2pq \sim 2q = 0.032$
Carriers/Affecteds = $2pq/q^2 \sim 2q/q^2 = 2/q = 125$

b. U.S. non-Ashkenazi Caucasian incidence: 1/112,000= q^2
$q = 0.003$, $p = 0.997$
$p^2 = 0.994$, $2pq \sim 2q = 0.006$, $q^2 = 0.000009$

Gene frequency = $q = 0.003$,
Carrier frequency ~ $2q = 0.006$
Carriers/Affecteds ~ $2/q = 667$

Ashkenazi carriers/non-Ashkenazi Caucasian carriers = 0.032/0.006 = 5.3

Example 2:
Huntington Disease (Autosomal dominant; p is disease allele):

Incidence: 1/10,000 = $p^2 + 2pq \sim 2pq \sim 2p$
$p = 1/20{,}000 = 0.00005$, $q = 1 - p = 0.99995$
Heterogygous Affecteds/Homozygous Affecteds ~ $2p/p^2 = 2/p = 40{,}000$

Determining if the Genotypes are in Hardy-Weinberg Equilibrium by a Chi-Square Goodness of Fit Test

(Using numbers from the example given under Subheading "Allele and Genotype Frequencies")

The MN Blood Group:
543 MM (phenotype M), 419 MN (phenotype MN), and 457 NN (phenotype N) individuals (total = 1419 individuals)

There are 543 *MM* individuals, which means there are 543 × 2 = 1086 *M* alleles being contributed. 419 *MN* individuals will contribute 419 *M* alleles and 419 *N* alleles. And there are 457 *NN* individuals contributing 457 × 2 = 914 *N* alleles.

Genotype	*Number of individuals*	*Genotypic frequencies*
MM (*phenotype M*)	543	543/1419 = 0.38
MN (*phenotype MN*)	419	419/1419 = 0.30
NN (*phenotype N*)	457	457/1419 = 0.32
Total	1419	1.0

Therefore, there are 1086 + 419 = 1505 *M* alleles total in the population and 419 + 914 = 1333 *N* alleles total in the population.

So, $p = f(M) = 1505/[2(1419)] = 0.53$ and $q = f(N) = 1333/[2(1419)] = 0.47 = 1 - p$.

A χ^2 test of "goodness of fit" is performed to determine if the population is in Hardy-Weinberg equilibrium proportions, given the values of p and q calculated above:

Phenotype	*Observed*	*Expected*
M	543	$p^2(1419) = (0.53)^2(1419) = 398.6$
MN	419	$2pq(1419) = 2(0.53)(0.47)\,1419 = 706.9$
N	457	$q^2(1419) = (0.47)^2(1419) = 313.4$

From subheading "Hardy-Weinberg and Linkage Equilibrium," the equation for the chi-square goodness of fit statistic, with one degree of freedom is:

$$\chi_1^2 = \sum_{genotypes} \frac{(Observed - Expected)^2}{Expected}$$

So, for this example,

$\chi_1{}^2 = (543 - 398.6)^2/398.6 + (419 - 706.9)^2/706.9 + (457 - 313.4)^2/313.4 = 235.36.$

Since the associated p-value < 0.0001, we reject the null hypothesis that the population is in Hardy-Weinberg equilibrium.

Allelic Linkage Disequilibrium Calculations for Two Simple, Biallelic Loci

Assume there are two biallelic loci, L_1 and L_2. L_1 has two alleles A and a and L_2 has two alleles B and b. The total sample size is 50 (n).

Hence, there are four possible haplotypes formed between the alleles of the two loci, L_1 and L_2: Ab, ab, AB and aB. The frequencies of these haplotypes are 20, 10, 5, and 15, respectively. Then

$f(Ab) = 20/50 = 0.40, f(\text{ab}) = 10/50 = 0.20, f(AB) = 5/50=0.10$ and $f(aB) = 15/50=0.30$ (Note: The sum of these frequencies equals one.) Then, $p_{L1} = f(A) = 20+ 5/50 = 0.50$, which means that $q_{L1} = 1 - pL_1 = f(a) = 1–0.50 = 0.50$

and that $p_{L2} = f(\text{B}) = 5 + 15/50 = 0.40$,

which means that $q_{L2} = 1 - p_{L2} = f(b) = 1 - 0.40 = 0.60$.

Calculate the four allelic linkage disequilibria for L_1 and L_2.

The equation for two-locus LD, $d(ij)$, given in the Hardy-Weinberg and Linkage Equilibrium section,

$d(ij) = f(ij) - f(i)f(j)$

$d(ij)$ = disequilibrium of allele i at locus 1, and allele j at locus 2;

$f(i)$ = frequency of allele i at locus 1, $f(j)$ = frequency of allele j at locus 2;

$f(ij)$ = frequency of haplotype ij (locus 1 and locus 2);

Hence for loci L_1 and L_2, we will calculate $d(Ab)$, $d(\text{ab})$, $d(AB)$ and $d(aB)$.

$d(Ab) = 0.40 - (0.50)(0.60) = 0.10$

$d(ab) = 0.20 - (0.50)(0.60) = -0.10$

$d(AB) = 0.10 - (0.50)(0.40) = -0.10$

$d(aB) = 0.30 - (0.50)(0.40) = 0.10$

Then, the two locus chi-square test, with one degree of freedom, is given by

$$\chi_{ij}^2 = \frac{n[d(ij)]^2}{f(i)\,[1 - f(i)]\,f(j)\,[1 - f(j)]}$$

So, for the haplotype Ab, formed from allele A from the first locus and allele b from the second locus,

$\chi^2_{Ab} = \{[50(0.10)^2]/ (0.50)(1 - 0.50)(0.60)(1 - 0.60)\} = 0.5/0.06 = 8.33.$

Since the associated p–value = 0.004, we reject the null hypothesis that the two loci are in linkage equilibrium at alleles A and b, i.e., reject the null hypothesis that $d(Ab) = 0$.

Suggested Reading

Hardy-Weinberg and Linkage Equilibrium

Hardy, G. H. (1908) Mendelian proportions in a mixed population, Science 28, 49–50.

Guo, S. W. and Thompson, E. A. (1992) Performing the exact test of Hardy-Weinberg proportion for multiple alleles, Biometrics 48, 361–372.

Hill, W. G. (1974) Disequilibrium among several linked neutral genes in finite populations. I. Mean changes in disequilibrium, Theor. Popul. Biol. 5, 366–392.

Hill, W.G. (1974) Estimation of linkage disequilibrium in randomly mating populations, Heredity 33, 229–239.

Li, C. C. (1955) Population Genetics, 1st ed., The University of Chicago Press, Chicago, IL.

Nei, M. (1987) Molecular Evolutionary Genetics, Columbia University Press, New York, NY.

Weinberg, W. (1908) Uber den Nachweis der Vererbung biem Menschen, Jh. Verein. f. vaterl. Naturk. Wurttemberg 64, 368–382.

Weinberg, W. (1909) Uber Verebungsgestze beim Menschen, Ztschr. Abst. U. Vererb. 1, 277–330.

Weir, B. S. (1996) Genetic Data Analysis II: Methods for Discrete Population Genetic Data, Sinauer Associates, Inc., Sunderland, MA.

Zaykin, D., Zhivotovsky, L., and Weir, B. S. (1995) Exact tests for association between alleles at arbitrary numbers of loci, Genetica 96, 169–178.

Darwin and Natural Selection

Darwin, C. (1859) On the Origin of Species, Murray, London, UK.

Darwin, C. (1871) The Descent of Man and Selection in Relation to Sex, D. Appleton and Company, New York, NY.

Fisher, R. A. (1930) The Genetical Theory of Natural Selection, 1st ed., Clarendon Press, Oxford, UK.

Haldane, J. B. S. (1932) The Causes of Evolution, Longmans and Green, London, UK.

Wright, S. (1931) Evolution in Mendelian populations, Genetics 16, 97–159.

Types of Variation

Ayala, F. J. and Kiger, J. A. (1984) Modern Genetics, 2nd ed., The Benjamin Cummings Publishing Company, Inc., San Francisco, CA.Watson, J. D. and Crick, F. H. C. (1953) Molecular structure of nucleic acids: A structure for Deoxyribase nucleic acid, Nature 4356, 737–738.

Hartl, D. L. and Clark, A. G. (1989) Principles of Population Genetics, 2nd ed., Sinauer Associates, Inc., Sunderland, MA.

Nei, M. (1987) Molecular Evolutionary Genetics, Columbia University Press, NY.

Vogel, F. and Motulsky, A. G. (1997) Human Genetics: Problems and Approaches, 3rd ed., Springer-Verlag, Berlin, Germany.

Weiss, K. M. (1993) Genetic Variation and Human Disease: Principles and Evolutionary Approaches, Cambridge University Press, Cambridge, MA.

Polymorphism

Harris, H. and Hopkinson, D. A. (1972) Average heterozygosity per locus in man: an estimate based on the incidence of enzyme polymorphisms, Ann. Hum. Genet. 36, 9–20.

Landsteiner, K. (1900) Zur Kenntnis der antifermentativen, lytischen und agglutinierenden Wirkungen des Blutserums und der Lymphe, Zentralbl. Bakteriol. 27, 357–362.

Genetic Drift and Migration

Wahlund, S. (1928) Zuzammensetzung von Populationen und Korrelation-Erscheinungen vom Standpunkt der Vererbungslehre ausbetrachtet, Hereditas 11, 65–106.

Weiss, K. M. (1993) Genetic Variation and Human Disease: Principles and Evolutionary Approaches, Cambridge University Press, Cambridge, MA.

Wright's Fixation Indices

Cockerham, C. C. (1969) Variance of gene frequencies, Evolution 23, 72–84.

Cockerham, C. C. (1973) Analyses of gene frequencies, Genetics 74, 679–700.

Wright, S. (1931) Evolution in Mendelian populations, Genetics 16, 97–159.

Wright, S. (1943) Isolation by distance, Genetics 28, 114–138.

Wright, S. (1951) The genetic structure of populations, Ann. Eugen. 15, 323–354.

Genetic Distance

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems, Ann. Eugen. 7, 179–188.

Mahalanobis, P. C. (1936) On the generalized distance in statistics, Proc. Natl. Inst. Sci. India 2, 49–55.

Nei, M. (1972) Genetic distance between populations, Am. Nat. 106, 283–292.

Sanghvi, L. D. (1953) Comparison of genetical and morphological methods for a study of biological differences, Am. J. Phys. Anthrop. 11, 385–404.

Other General References of Interest

Cavalli-Sforza, L. L. and Bodmer, W. F. (1971) The Genetics of Human Populations, W.H. Freeman and Company, New York, NY.

Crow, J. F. (1986) Basic Concepts in Population, Quantitative, and Evolutionary Genetics, W.H. Freeman and Company, New York, NY.

Crow, J. F. and Kimura, M. (1970) An Introduction to Population Genetics Theory, Harper & Row, NY.

Falconer, D. S. and Mackay, T. F. C. (1996) Introduction to Quantitative Genetics, 4th ed., Longman Group Ltd., Essex, UK.

Felsenstein, J. (1981) Bibliography of Theoretical Population Genetics, Dowden, Hutchinson, and Ross, Stroudsburg, PA.

Hartl, D. L. (1980) Principles of Population Genetics, Sinauer Associates, Sunderland, MA.

Hartl, D. L. (1981) A Primer of Population Genetics, Sinauer Associates, Sunderland, MA.

Kimura, M. (1983) The Neutral Theory of Molecular Evolution, Cambridge University Press, Cambridge, MA.

Kimura, M. (1994) Population Genetics, Molecular Evolution, and the Neutral Theory, Selected Papers, (Takahata, N., ed.), The University of Chicago Press, Chicago, IL.

Lewontin, R. C. (1974) The Genetic Basis of Evolutionary Change, Columbia University Press, New York, NY.

Suzuki, D. T. Griffiths, A. J. F., and Lewontin, R. C. (1981) An Introduction to Genetic Analysis, 2nd ed., W.H. Freeman and Company, San Francisco, CA.

Winter, P. C., Hickey, G. I., and Fletcher, H. L. (1998) Instant Notes in Genetics, BIOS Scientific Publishers Limited, Oxford, UK.

Wright, S. (1968) Evolution and the Genetics of Populations, vol. 1, Genetics and Biometric Foundations, University of Chicago Press, Chicago, IL.

Wright, S. (1969) Evolution and the Genetics of Populations, vol. 2, The Theory of Gene Frequencies, University of Chicago Press, Chicago, IL.

Wright, S. (1970) Evolution and the Genetics of Populations, vol. 3,The Experimental Results and Evolutionary Deductions, University of Chicago Press, Chicago, IL.

Part III

The UNIX Operating System

A. Basics and Installation

13. Introduction to UNIX for Biologists
14. Installation of the Sun Solaris™ Operating Environment
15. Sun System Administration

13 Introduction to UNIX for Biologists

David D. Womble

Introduction

Much of the bioinformatics software described in the other chapters of this book (for example, the GCG or EMBOSS programs; e.g., *see* Chapters 18, 19, and 24) is installed on computers that have a UNIX operating system, such as Sun Solaris or Linux. In this chapter, the reader will receive an introduction to using computers with the UNIX operating system. This is not a comprehensive course in UNIX, but is written for general users who want to work with UNIX computers and UNIX software. Note that the perspective used for this chapter is that of a biologist, not a computer scientist. After a general introduction, the reader will be presented with examples and tutorials that will demonstrate several tasks, such as file management or text editing, on UNIX computers using either the command line or the graphical windows interface. A table of simple but useful UNIX commands is included in Appendix 3.

General Introduction to UNIX

A computer operating system is the program that allows a computer user to access the various resources, or parts, of the computer, such as memory, hard disks, or the display screen. It also provides the foundation for the user to install and run software or programs, to do work with the computer.

UNIX is a computer operating system that allows a computer to be shared providing secure simultaneous access for multiple users. It is a multitasking operating system, which means that a UNIX computer can carry out multiple tasks, such as commands or programs, all at the same time. UNIX allows the sharing of resources such as files and hardware across a network. Users can connect to a UNIX computer from anywhere on the network and operate the computer from a remote location.

Because UNIX is a multi-user operating system, a UNIX computer can be used by a group of users, either at different times or all at the same time. Each UNIX user has an account with a username and password that are used to *login* to the computer. Users can sit down in front of the computer and login using the keyboard, mouse, and monitor that are attached to the computer. This is called *logging onto the console*. Alternatively, users can connect to the UNIX computer from remote locations using terminals or terminal software on other computers connected via a network, such as Ethernet.

Note: The author has no affiliation with any of the software sources listed in this chapter.

There are several different versions of UNIX, and most of the larger computer vendors, such as Sun, IBM, and Compaq, ship their computers with their own versions of UNIX. Sun Solaris, a widely used version of UNIX, is one example that runs on computers sold by Sun Microsystems. Installation and systems administration of the Sun Solaris UNIX operating system are described in Chapters 14 and 15. Linux, an open source UNIX-like operating system, can be installed on personal computers with an Intel X86 processor. Most of the information in this chapter will be generally applicable to most any version of UNIX. However, Solaris will be the version that is used for the examples that follow.

On a UNIX computer, commands are used to instruct the computer to perform tasks. There is more than one way to issue commands. Most UNIX computers have a graphical windows interface that allows the user to use a mouse to select and execute various actions or to start programs. However, the most basic means to operate a UNIX computer is to use the command line to type the name of a command, followed by pressing the **return** or **enter** key to execute the command. The graphical windows interface and the command line of a UNIX computer can be used either at the console or remotely through the network.

UNIX commands instruct the computer to do something. The commands act on *input*, e.g., what the user types or is a file stored on the computer's hard disk. The result of the command is the *output*. The output is either displayed on the screen or is saved in a file.

Switches are additions to the UNIX command that change the way it performs its actions. Without a switch, the command will act in its *default* or normal mode. With a switch, it will act in a different, usually more complex, way. These switches apply to both the graphical interface and the command line. In the graphical interface, switches are often activated by using the mouse to check boxes next to various options. On the command line, the options are typed after the command before pressing the **return** key.

In the examples and tutorials that follow, it is assumed that the users will be connecting to and running programs on a UNIX computer called *genetics*, and that the genetics server is running the Sun Solaris operating system and is connected to the Internet. Other UNIX systems are similar, but some details may vary.

Logging on to a UNIX Computer

The first step is to have a user's account set up on the UNIX computer. The UNIX administrator, who has special privileges to manage the computer, manages the accounts including the creation of new accounts. Systems administration for the Sun Solaris operating system is described in Chapters 15 and 17. Once a user's account is set up, the user can either login at the console (i.e., at the computer itself) or from a remote location in the network.

The most basic method to connect to a UNIX computer from a remote location (such as from an office down the hall, from a laboratory across the campus, or from a hotel room across the country) is a program called *telnet*. Telnet is a computer program that allows the user to connect their local computer, such a personal computer, to the UNIX computer through the network to control the UNIX computer and run programs. A basic telnet program is currently included with most computer operating systems, and there are many examples of freeware or shareware telnet (or terminal) programs available. The user starts the telnet program like QVT/Term on the local computer, then connects to the remote server by typing **open genetics**.

In this case, the name of the remote UNIX computer is *genetics*, but it would most likely be in the format of an Internet address such as *genetics.myuniversity.edu* or something similar.

Once the connection is established, the login screen would appear, similar to:

login:

The user would then enter their username (e.g., janeuser), after which would appear:

Password:

The user would then type their password. For security reasons, the password itself does not appear on the screen. After entering the password, a UNIX command prompt would appear, similar to this:

genetics%

The UNIX command prompt is often the name of the computer followed by the percent sign (%). The command prompt will be at the left side of the screen on what is referred to as the command line and, as one would expect, commands are typed on that line to operate the computer.

Although any basic telnet program will work as described earlier, the author prefers to use a shareware program called QVT/Term, which can be downloaded from the Internet (*see* Website: http://www.qpc.com). QVT/Term is particularly good for connecting to UNIX servers running programs such as the GCG package (*see* Chapters 18, 19, 28, and 33), since it is easy to print the results from the GCG programs through the network. Printing works well for GCG programs operated either from the command line or from the graphical windows interface for GCG called SeqLab, both of which are described in Chapters 18, 19, 28, and 33. The QVT/Term telnet screen is shown in Fig. 1.

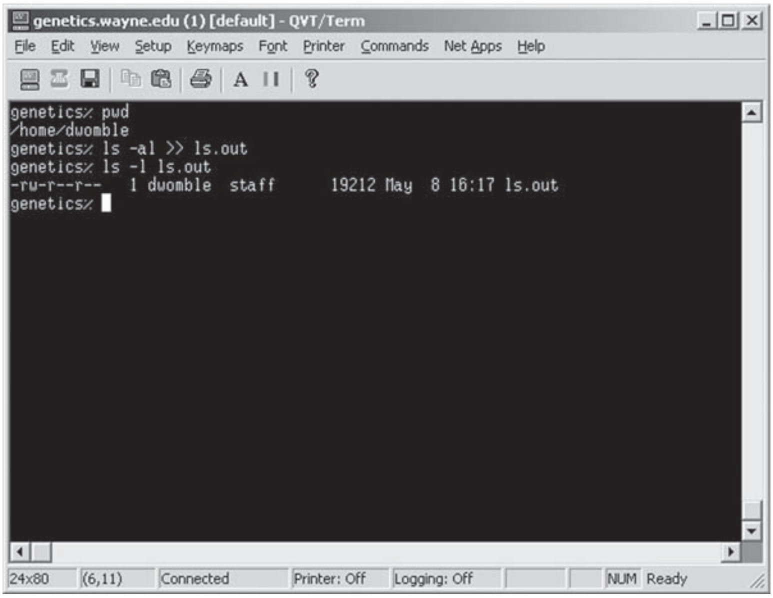
Other than using telnet, the login procedure described earlier also applies to logging onto the console. Either way, once the UNIX command prompt is displayed, the user can type the names of UNIX commands or programs.

Computer Files

In general, most operations that computers carry out are done with files. Files are pieces of information that are stored in directories (or folders) on the UNIX computer's hard disk(s). Files have names such as *my_dna_sequence.txt*. Files can contain data, such as a DNA sequence, or they can be programs, such as one of the GCG programs that will analyze the data in a sequence file. The following examples and tutorials describe how to manipulate files stored on the UNIX computer's hard disk(s).

Home Directory

After logging onto a UNIX computer, the user will be located in a folder on the computer's hard disk called the user's *home directory*. On a multi-user UNIX computer, each user will have their own home directory. The user is allowed to make changes to files located in their home directory (and subdirectories thereof), such as creating new files, editing existing files, or copying, renaming, or deleting existing files.


genetics.wayne.edu (1) [default] - QVT/Term
File Edit View Setup Keymaps Font Printer Commands Net Apps Help
24x80 (6,11) Connected Printer: Off Logging: Off NUM Ready

```
genetics% pwd
/home/dwomble
genetics% ls -al >> ls.out
genetics% ls -l ls.out
-rw-r--r--   1 dwomble  staff      19212 May  8 16:17 ls.out
genetics% 
```

Fig. 1. The QVT/Term Telnet Screen Showing the UNIX Command Prompt.

Usually, users will not be able to make changes to files except those in their own home directories, so that one user cannot change files in another user's home directory. An exception is the *superuser*. The superuser is usually the owner or administrator of the UNIX computer who has special permissions to change all files on the computer, including those of the operating system itself or of the programs that are installed in the public areas of the computer for all users to run. Ordinary users cannot usually change operating system files, the publicly installed programs, or even other ordinary users' files.

Using the Command Line

The following examples and tutorials will demonstrate how to use the command line to operate on files and directories on a UNIX computer. The examples assume the user has logged into a UNIX server and has the **genetics%** command prompt available.

To get the most out of these examples and tutorials, they should be followed in sequence, as presented here. Please note that UNIX is case-sensitive, so it is important to type the commands exactly as indicated. For example, on a UNIX computer, the filenames *myfile* and *Myfile* indicate two different files, both of which may be located in the same folder.

Where Am I? (**pwd** command)

To see the name of the current directory in which the user is located, the user types: **pwd** and presses the **return** key. The result, or output, is the name of the current working directory, which will be displayed on the screen as:
/home/joeuser

That indicates that the user, whose username is *joeuser*, is currently located in their home directory. Their home directory is a folder named *joeuser*, and that folder is a subfolder of the main folder named *home*. The full directory path to their home directory is */home/joeuser*.

The details of the names of home directories and folders may vary depending on how the UNIX administrator has set up the user accounts, but the results shown earlier are typical.

In this first example of using the command line, the UNIX command was the **pwd** command, and the output was the name of the current working directory. This is typical of how commands are executed, and how the output is displayed on the screen. The other examples that follow are similar.

What Files Are In My Directory? (ls command)

To see a list of the files that are located in the user's current working directory, the user types the list files command **ls**. The result is an alphabetical list of the names of the files in the current working directory, similar to:

```
file1.txt   file2.txt
```

Adding Switches to Change How a Command Operates

In the earlier example, the **ls** command was used in its default or normal mode. Most UNIX commands have switches that change the way the command operates. In the following examples, switches are added to the **ls** command to change the kind of information listed about the files in the current working directory.

To see more information about the files, such as their size and modification dates, the user would add the *longform* switch **-l** to the **ls** command. Switches are added to a UNIX command by typing a minus sign before the switch, **ls -l**. The user types the command, types a space, types a minus sign, then, without typing a space, types the switch, then presses the **return** key. The result is an alphabetical list of the files with all their longform information:

```
-rw-r--r--   1   joeuser   staff        10 May   7   15:51   file1.txt
-rw-r--r--   1   joeuser   staff        17 May   7   15:51   file2.txt
```

Although it is beyond the scope of this chapter to describe in detail all information, the basic pieces are, from right to left: the name of the file, the time and date it was last modified, the size in bytes, the name of the group to which the file's owner belongs, the username of the owner, the number of links to the file, and the read, write, and execute permissions associated with the file.

Hidden Files (dot files)

To see a list of all the files in the current working directory, including hidden (or *dot* files), the user would add the *all* switch **-a** to the **ls** command, **ls -a**. The result would be a list of all the files in the current working directory, including those filenames that begin with a period, or dot, for example:

```
./   ../   .dotfile1    .dotfile2    file1.txt    file2.txt
```

Dot files located in a user's home directory are used to configure the operation of programs that the user runs. For example, files named *.cshrc* and *.login* are read by the operating system each time the user logs in. These files are required, so they should not be deleted. However, experienced users can edit the dot files to customize the way various programs operate according to their preferences.

Adding Multiple Switches to the Command Line

Switches can be combined in one command. An example would be to use the all and longform switches together with the **ls** command. That can be done either together or separately, such as:

```
\.ls -al
ls -a -l
```

The output might look similar to this:

```
drwxr-xr-x   2  joeuser  staff          512  May  7  15:54  ./
drwxr-xr-x  42  joeuser  staff         3072  May  7  15:53  ../
-rw-r--r--   1  joeuser  staff            0  May  7  15:54  .dotfile1
-rw-r--r--   1  joeuser  staff            0  May  7  15:54  .dotfile2
-rw-r--r--   1  joeuser  staff           10  May  7  15:51  file1.txt
-rw-r--r--   1  joeuser  staff           17  May  7  15:51  file2.txt
```

The UNIX Manual (**man** command)

In most cases, it does not matter in which order the switches are typed after a command. However, to see the details of what switches are available and how to use them, the user should consult the online UNIX manual by typing the **man** command. The user types the **man** command followed the name of the command of interest: **man ls**. That will display the manual about the **ls** command, which includes a list of all switches available, what they do, and how to use them. To see more of the manual, i.e., to advance to the next page, press the **Spacebar** key. To move to the previous page, press the **b** key. To quit reading the manual, press the **q** key.

Adding Arguments to a Command

In the earlier example of using the **man** command, some additional input was typed on the command line following the **man** command, i.e., the name of the command (**ls**) for which the manual was wanted. This additional input is called an *argument*. Some commands, such as the **man** command, require an argument on the command line. For other commands, such as the **ls** command, an argument may be optional. As an example, to list the files in the user's home directory, the user could type the whole name of the directory after the **ls** command: **ls /home/joeuser** . That would list all the files in the folder named */home/joeuser*.

Redirecting Output to a File (> symbol)

In the earlier examples for the **ls** command, the results (output) appeared on the screen. The results can be saved to a file instead. This is called *redirecting output to a file*. This is done by using the greater than symbol "**>**", the user types:

```
ls -al > ls.out
```

That will cause the results of the **ls -al** command to be saved into a file called *ls.out*. The ">" (greater than) symbol redirects the output of the command into the file. If two greater than symbols are used, like this: **ls -al >> ls.out** , the output from the **ls -al** command is appended onto the end of the file *ls.out*, saving the existing data in that file.

How Can I See What's In a File? (**more** and **cat** commands)

Use the **more** command to see the contents of a file, type:

more ls.out.

The **more** command displays the contents of a file one page at a time, e.g., the file named *ls.out* created in the previous section. To see more pages (if any) press the **spacebar** key. To move back a page, press the **b** key. To quit reading the file, press the **q** key.

Use the **cat** command to display the entire file contents scrolled across the screen. Type:

cat ls.out.

Using Multiple Commands on One Line (| symbol)

The **cat** command is not very useful by itself, but can be used in combination with other commands. To send the results from one command into another command, the user types the pipe symbol "**|**" like this: **cat ls.out | more**.

In this example, the results (output) of the **cat** command, instead of being displayed on the screen, are sent (piped) into the **more** command, i.e., the output of the **cat** command is used as input for the **more** command.

This example might seem a bit facile, since the **more** command could have been used directly, but it illustrates a powerful capability of the UNIX command line. Experienced users can write scripts that will automatically process large numbers of files or large amounts of data by sending the results of one command to be processed by other commands down the line.

How Do I Copy a File? (**cp** command)

Use the **copy** command to copy a file, type: **cp ls.out ls.copy**.

This will make a copy of the *ls.out* file and name it *ls.copy*. Note that in this case, two arguments are added to the **cp** command, the name of the file to copy, and the name of the destination file. The result is two files, the original and the copy. Other than its filename, it is identical to the original.

How Do I Rename a File? (**mv** command)

Use the **move** command to move a file to a new name, type: **mv ls.copy junk**. That will change the name of the *ls.copy* file to *junk*.

How Do I Delete a File? (**rm** command)

Use the **remove** command to delete a file, type: **rm junk**. That will delete the file named *junk*. Note that there is no **undelete** command in UNIX.

How Do I Create a New Folder? (**mkdir** command)

If the user is located in their home directory, the user can use the **make directory** command to create a new subfolder by typing:

```
mkdir Newfolder
```

That will create a new folder named *Newfolder*, and in these examples, it would have a complete name of:

/home/janeuser/Newfolder

How Do I Change to a Different Folder? (**cd** command)

To go to a different directory, use the **change directory** (**cd**) command to change from the current working directory to some other directory, for example type:

```
cd Newfolder
```

or

```
cd /home/janeuser/Newfolder
```

Either command line may take the user to the new folder. The first assumes that the user is already in their home directory. The second will change into the new folder from anywhere, no matter what the current working directory.

To change back to the user's home directory from any other directory, the user simply types the **cd** command without an argument.

How Do I Delete a Folder? (**rmdir** command)

Use the remove directory (**rmdir**) command to delete a folder, such as the one created in a previous section named *Newfolder*. The folder must not contain any files, i.e., it must be empty before it can be removed. Use the **cd** and **rm** commands to change into and delete the files in the folder. Then use the **cd** command to change to a different folder, such as the user's home directory. Then delete the folder by typing:

```
rmdir Newfolder
```

How Do I Edit a File?

Any text file in a user's home directory can be edited with a text editor. Which editor is used depends on which text editing programs are installed on the UNIX computer. Most UNIX computers have a standard text editor called *vi* that can be used to edit any text file. The vi editor is powerful and has many features, but it is also fairly difficult for a beginner to use.

An easy-to-use text editor that may be installed on many UNIX computers is called *pico*. The pico text editor is what is used to edit email messages in the popular pine email program—pico stands for pine composer. It is beyond the scope of this chapter to explain how to use either vi or pico. However, both can be started from the command line, similar to this example:

```
pico textfile.txt
```

This command would start the pico editor and open up a file named *textfile.txt.* If a file in the current working directory by that name already exists, it will open that file. If no such file exists, it will create a new file with that name. Once the file is opened, it can be edited by typing new text, deleting text, cutting and pasting. Other text editing procedures can also be applied. When finished editing, the user saves the edited file and quits the program.

How Do I Run the GCG Programs From the Command Line?

The main reason for logging onto a UNIX computer is to run programs to do useful work. One example is the GCG sequence analysis programs. These can be run by several different means, including the command line. Of course, the systems administrator must have already installed the GCG programs and made them publicly available to the ordinary users.

To start the GCG programs from the command line, the user usually types **gcg** and presses the **return** key. This initializes the GCG programs, so that the user can then type the names of GCG programs, including adding switches, just as if they were UNIX commands. To run the restriction mapping program, the user would type:

```
map
map mysequence.seq
```

The first method simply starts the mapping program, which will then ask the user for the name of the sequence data file to process. The second uses the name of the sequence file as an argument on the command line, and the map program will then run on that particular data file.

Command line use of the GCG programs, and many other kinds of programs, are described in Chapters 18 and 19.

How Do I Logout From a UNIX Computer? (**exit** command)

To courteously logout from a UNIX computer, or from a telnet session connected to a UNIX computer, type the **exit** command. This will shut down the login and exit the user from the UNIX computer. One should never just close the telnet program while connected to a remote UNIX computer, as the UNIX host might not realize the connection is finished and keep resources open that could better be used to serve other users.

What Other UNIX Commands are Available?

A table listing of a few simple and useful UNIX commands is included in Appendix 3. Many other commands are available, but those in the appendix are generally useful for the beginner. For those wishing to acquire more detailed information, there are many books devoted to using UNIX for users of all skill levels, including entire books dedicated to using the vi text editor. Additional information for UNIX systems administrators is presented (*see* Chapter 15).

Using the Graphical Windows Interface: X Windows

UNIX computers use a graphical windows interface called *X Windows*. A UNIX computer can display its X Windows on its own local display screen or on the display screen of some other remote computer connected via the network. Having the X Windows displayed remotely allows the remote user to control the UNIX computer through the graphical interface, using the mouse to select commands, programs, or files on which to operate. This is an alternative to using telnet and the command line, which some users may prefer. In the examples and tutorials that follow, X Windows will be used to operate files and folders on a remote UNIX computer, similar to the operations that were executed from the command line in the earlier sections of this chapter.

The key to having the remote computer display the X Windows locally is to have an X Windows server program installed on the local computer. Most UNIX (or Linux) computers have some kind of X Windows server installed by default. To display X Windows on an IBM personal computer (PC) or an Apple Macintosh (Mac), a third party X Windows server program is required. The author uses a commercial X Windows server program called *XWin-32*. There are more complex (and more expensive) programs available, but XWin-32 works well with the GCG or Staden programs (*see* Chapters 20, 24, 28, and 33). XWin-32 is available for download from the Internet (*see* Website: http://www.starnet.com/products/).

Setting Up Two Computers to Use X Windows

In the following examples, the user is assumed to be using a local computer that can connect to a remote UNIX computer through the network. The local computer can also be a UNIX or Linux computer, but it can also be a PC or Mac with X server software installed. The author prefers to use QVT/Term to connect and login to the remote UNIX computer, and start the X programs by typing them on the command line. The X Windows from the remote UNIX computer will be displayed on the user's local computer display screen.

There are three steps required to use X Windows: 1) set up the local computer to receive X Windows from a remote computer; 2) tell the remote UNIX computer where to display its X Windows; 3) run the X Windows programs on the remote computer.

For the first step, the user must install an X server program on their PC or Mac, or be using a UNIX or Linux computer with built-in X server software. On a local PC or Mac, the user then starts the X server program and leaves it running, ready to accept X Windows from a remote computer. From the console, often on a local UNIX or Linux computer, the user types the **xhost** command, example:

```
xhost +
```

That toggles on the *xhost* function. For more security, the user can add the Internet (IP) address of the remote computer to the **xhost** command line. This will restrict X displays to that particular computer. To turn off the xhost feature, the user types:

```
xhost –
```

The details of using the **xhost** command may vary, but are contained in the manual pages that are part of the operating system. They can be viewed by entering the **man xhost** command.

For the second step, the user must tell the remote UNIX computer the Internet (IP) address of their local computer. So, it is necessary to find out what that local IP address is. That usually can be done by looking in the network configuration control panel of the local computer, but it may require help from the local network administrator.

The easiest way to use X Windows programs on the remote computer is to first login to the remote computer using telnet as described earlier in this chapter. Once logged on, the user can tell the remote UNIX computer where to display its X Windows by setting the DISPLAY environment variable by typing on the command line:

```
setenv DISPLAY my.ip.address:0
```

Substitute the local PC's IP address, and don't forget the **:0** at the end. That sets an *environment variable* called *DISPLAY* to the value of the PC's IP address. If the user always connects from a PC with the same IP address, the user can put that line in their *.login* file.

Alternatively, one can set up the X server program on the PC to include all the display information along with the X program that will be remotely executed. Although that can be convenient, it does require extra effort.

It may also be necessary to set another environment variable, which can be done by adding this line to the user's .cshrc file:

```
setenv LD_LIBRARY_PATH /usr/openwin/lib
```

The details of what needs to be on that line may vary with the particular UNIX computer, so the user should consult with the UNIX systems administrator.

Running X Programs on the Remote UNIX Computer

Having accomplished the first and second steps needed to set up two computers to run X Windows as described earlier in this chapter, a user can type the name of any X program on the command line (of the remote computer), for example:

```
filemgr &
```

That will start the Sun graphical file manager program, which will display on the user's local computer screen. The user can then use their mouse to accomplish the tasks outlined in the examples and tutorials. The **&** on the end of the command line causes the *filemgr* program to run in the background, so that other commands can be typed from the command prompt.

Alternatively, the user could build an executable command icon on their local computer screen using the X server software. The icon would include the name of the remote host, the remote execution method, and the following command line:

```
/usr/openwin/bin/filemgr -display your.ip.address:0 &
```

Selecting this single icon will accomplish everything described above. Different icons can be created for each X program that the user wishes to run on the remote computer.

Using the Graphical File Manager

The Sun graphical file manager window is shown in Fig. 2. When *filemgr* starts, the user is located in a folder on the hard disk called their home directory. Their home directory is usually something like */home/joeuser*.

How Can I See Information About My Files?

There are several different ways to display information in the File Manager window. For example, using the mouse, right click on the **View** menu, then left click on **List by Name**. To see the hidden files as well, right click the **Edit** menu, left click

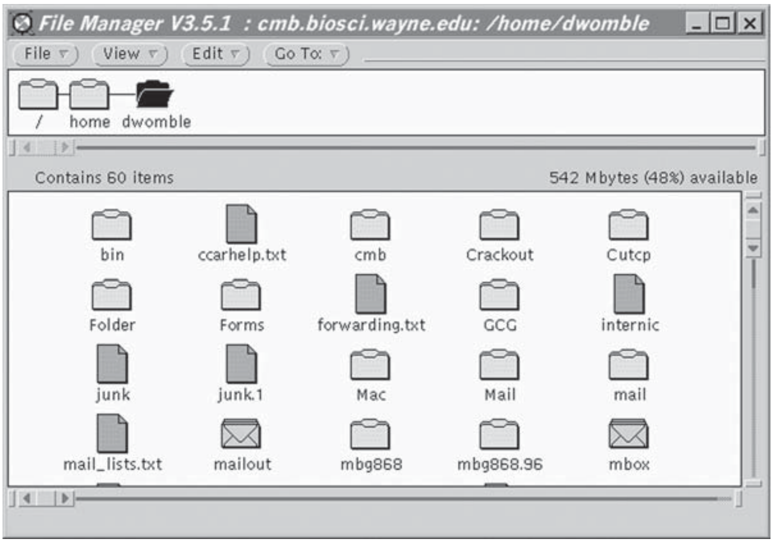


Fig. 2. The Sun Graphical File Manager Window.

on **Properties**, then in the new box, right click the **Category** button, then left click on **Current Folder Settings**, and click on **Show** next to Hidden Files, click the **X** in the upper right, then click on **Apply Changes** to apply the changes you just made. The user should see all the longform information about all their files in the File Manager window. Reverse the steps and click on **Icon by Name** under the **View** menu to return to the original view.

How Can I See What's In a File?

Using the mouse, left click once on a file to select it, then right click on the **File** menu, then left click on **Open in Editor**. The file will open in the X graphical text editor. Use the mouse to scroll through the contents. Left click the **X** in the upper right to close the file.

How Do I Copy a File?

Left click once on a file to select it, right click on the **Edit** menu, left click on **Copy**, right click on the **Edit** menu, left click on **Paste**, then click on **Alter name**. Type in a different name if desired.

How Do I Edit a File?

Left click once on a file to select it, then right click on the **File** menu, then left click on **Open in Editor**. Or if it is a plain text file, just double click on the file. Edit as you like, using the mouse to select text to cut, copy, or paste. To save the changes, right click on **File**, then left click on **Save** or **Save as**. Left click the **X** in the upper right to close the editor.

Alternatively, from the telnet and the UNIX command prompt, type **textedit &** to start the text editor program, independently of the File Manager. After the editor opens, right click on the **File** menu, left click **Open**, left click on the file to be edited, and left click on **Open**. Left click the **X** in the upper right to close the editor.

How Do I Rename a File?

Left click on the name of the file, then type the new name and press **enter**.

How do I Delete a File?

Left click once on a file to select it, right click on the **Edit** menu, left click on **Delete**. Note that unlike deleting a file from the command line, the deleted file is saved into the Waste Basket, from which it could be later retrieved.

How Do I Create a New Folder?

Right click on the **File** menu, left click on **Create Folder**, then type the name for the new folder and press **enter**. To create a new file, click on **Create Document**.

How Do I Change to Other Folders?

Double click on the folder you want to change to, or type the name of the folder on the *Go to* line. Alternatively, right click on the **Go to** menu, then click on one of the folders listed, or on **Home** directory.

How Do I Exit From the File Manger Window?

Left click on the **X** in the upper right, click on **Quit File Manager**, then click on **Yes** to empty the files from the waste.

How Do I Start the GCG SeqLab Interface?

Assuming that the GCG programs are installed on the remote UNIX computer, and that the user is connected via telnet and X Windows is configured as described, you first initialize the GCG programs from the command line by typing:

```
gcg
```

Once the GCG programs are activated, start the SeqLab program by typing:

```
seqlab &
```

That will start the SeqLab interface and display the SeqLab window on the user's local computer screen. The SeqLab Window is shown in Fig. 3.

The GCG SeqLab window allows the user to operate the GCG suite of sequence analysis programs by using the mouse to select sequences and programs. Using the GCG programs from both the command line and from SeqLab are described in subsequent Chapters 18, 19, 28, and 33.

Many users find using the X Windows interface easier than using the command line. However, experienced users may prefer to use the command line for several important reasons. For example, from the command line, it is possible to run programs reiteratively with the use of switches. This permits a high degree of automation that is not usually possible when running programs one by one from a graphical interface, or from the web.

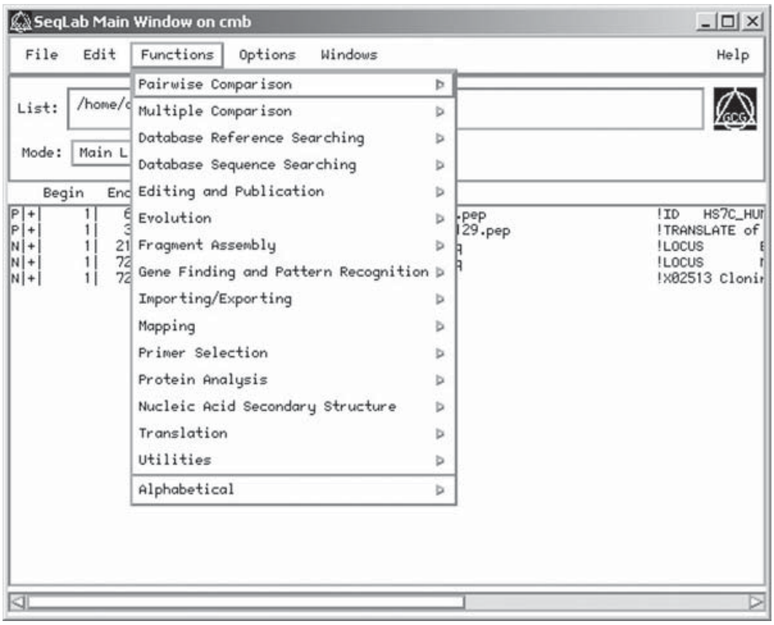


Fig. 3. The GCG SeqLab X Windows Interface.

Other Resources

As mentioned earlier, there are many books devoted to using UNIX available, for readers of all skill levels. In addition, for users of the GCG programs, the GCG manuals contain good basic introductions to using UNIX both from the command line and with X Windows, with particular emphasis on how to use UNIX for the purposes of running the GCG programs. The GCG manuals are usually available online at most installations, via the Internet in either HTML or in PDF format, or both, so the users can look up or print the information that is of interest to them.

14 Installation of the Sun Solaris™ Operating Environment

Bryon Campbell

Introduction

We will focus on the installation tasks for Solaris 8 on Intel® and SPARC® processor systems. Information for Sun Solaris™ running on SPARC® architecture is detailed at Website: http://www.sun.com/solaris and for Intel at Website: http://www.sun.com/intel. Additional documentation for Sun Solaris™ can be found at Website: http://docs.sun.com. As always, your User and System Administration Guides that are part of any Solaris purchase or locally available UNIX/Solaris™ experts are excellent resources.

Sun Solaris™ System Installation

Prior to installing the Solaris8 Operating Environment, fundamental system design decisions are required to ensure the installation process will be smooth.

- Define system use.
- Determine the SPARC® or Intel® hardware platform you will use.
- Determine how much disk space will be needed and how it will be partitioned.
- Determine how your system will be connected to your network.

System Use

Before you proceed you must determine how the system will be used. If the system will be used to develop applications, you will need plenty of processor speed and physical memory. If your intent is to have the system available for multiple users, you will need to plan for the correct amount of disk space to support their storage needs. If the system is also used as an application server, you will need to adjust the system hardware to meet the computational needs of the applications. Your local hardware vendor or Information Technology support organization can also assist you.

Hardware Platform

The Solaris™ operating system was first supported on the SPARC® platform. The first SPARC® systems were the SPARC® 5, 10, and 20. These systems were eventually replaced by faster UltraSPARC® systems that achieved speeds of 400 MHz and

Table 1
Software Distribution

Distribution	*Size in MB*
End User Support (smallest)	363 + swap space
Developer Support	801 + swap space
Entire Distribution	857 + swap space
Entire Distribution and OEM	900 + swap space

greater. It wasn't until Solaris™ version 2.1 that the Intel® architecture was supported, so that one could now install Solaris™ in dual boot mode on their Intel-based system. Potential Solaris Intel® x86 users should verify ISA, EISA, and PCI hardware compatibility in the Solaris 8 Operating Environment (*see* Website: http://access1.sun.com/drivers/hcl/hcl.html).

Disk Configuration

Before you decide on a disk-partition scheme, you will first need to decide the distributions that you will be installing. There are four distributions to choose from, *see* Table 1 for disk space recommendations.

The smallest distribution will work well for environments that do not require applications development. If you plan to develop applications it would be wise to install the entire distribution with the OEM (original equipment manager). Keep in mind that disk space is relatively inexpensive.

When partitioning the disk, there are five common partitions that will be created. These partitions should be a minimum of 300 MB and could extend in size to 1 GB +, depending on need. The partitions are as follows:

- **/ –** This is the root partition. This is also the top-level directory. Root is where all files that support the OS reside.
- **/home** The partition contains the user files. Directories below the home level will contain user specific information for users with accounts on the system.
- **/opt** This location contains the optional applications. Depending on how many optional applications are stored, this area could require substantial disk space.
- **/usr** This partition contains the utilities used in the Solaris operating system. The /bin and /sbin directories are where most of these utilities are located.
- **/var** This directory contains the log files, mail files, and print files. If your system is providing print services, you will need to ensure this partition is quite large so that it may accomodate the spool files associated with printing.
- **/etc** This serves as the location for machine specific configuration files.

As emphasized, disk space has become relatively inexpensive. You should buy a large disk, containing 10's of GB's, to minimize future growth issues as you develop and or host additional users.

Network Connectivity

As you install the Solaris™ Operating Environment, you will be asked to provide network configuration parameters to complete the installation. The basic parameters that will be needed are:

- Hostname: The name you give to your system.
- IP Address: The Internet Protocol (IP) address that provides network access to your machine.
- Domain Name: The name that uniquely identifies your system on the Internet. The fully qualified format is *hostname.domainname.com*.
- Subnet Mask: In environments with multiple networks or systems, the *subnet mask* is required to connect to other systems outside of your subnet. Your network administrator can provide detailed information.
- Default Router (Gateway): This provides the network path to systems outside of your network or subnet. Your network administrator can provide detailed information.

Note: If your site utilizes the Dynamic Host Configuration Protocol (DHCP), your system would automatically configure everything except the *hostname*.

Methods for Installation

There are three options for installing the Solaris™ Operating Environment: Solaris Interactive Installation, Jumpstart, and WebStart.

The Solaris™ Interactive installation can be completed on both SPARC® and Intel® platforms. There are two modes for installation, graphical and text mode. Availability of these modes will depend on the capability of the monitor and the keyboard connected to the system you are installing. The installation procedures are identified for both platforms.

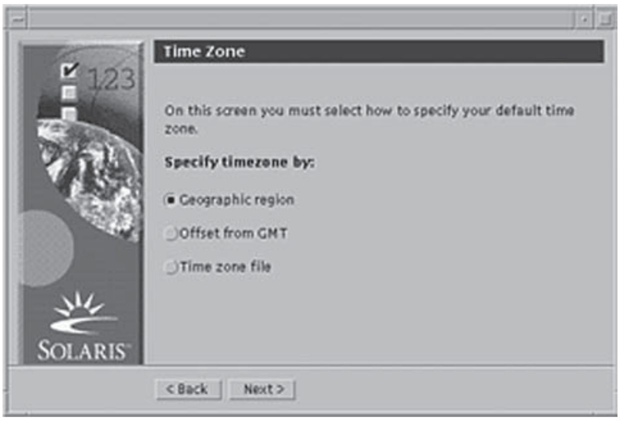
The Jumpstart installation is typically used for large Solaris™ installations, where another server has been pre-configured to install Solaris™ on new machines. You will want to work with your local system administrator to see if this is an option for you.

The WebStart installation provides a Java-based interface that walks you step by step through the installation process. The system requirements are higher than the minimum hardware requirements set forth by Sun. The Sun installation manual can provide additional WebStart information. A sample WebStart installation is presented below.

Installing the Solaris Operating Environment Using WebStart

There are two ways to accomplish a WebStart Installation. You can use the graphical user interface or the command line interface. The graphical installation requires a local or remote CD-ROM drive, network connection, keyboard, frame buffer, and graphics monitor.

With the command line interface, you have the same requirements except that a graphics monitor and frame buffer are not required. The sequence of instructions for both the graphical and command line interface installation are very similar. Figures 1 and 2 show sample screen shots.



Time Zone

On this screen you must select how to specify your default time zone.

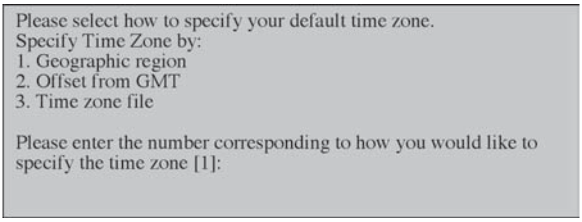
Specify timezone by:

● Geographic region

○ Offset from GMT

○ Time zone file

Fig. 1. Sample screen shot of the Graphical User Interface (GUI) Time Zone Dialog Box.



```
Please select how to specify your default time zone.
Specify Time Zone by:
1. Geographic region
2. Offset from GMT
3. Time zone file

Please enter the number corresponding to how you would like to
specify the time zone [1]:
```

Fig. 2. Sample screen shot of Command-Line Interface (CLI) Time Zone Dialog Box.

Pre-Installation Steps

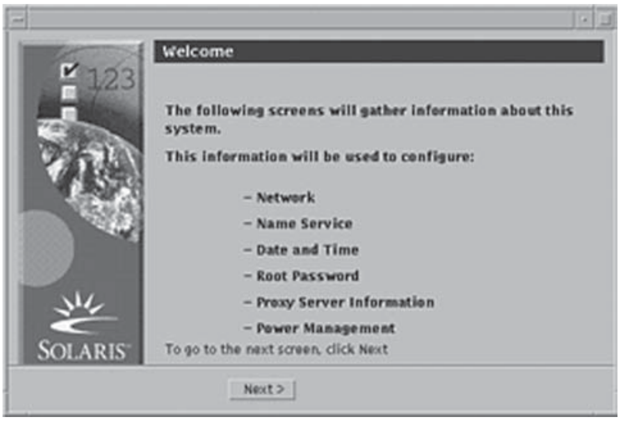
To determine how to install Solaris™, you will need to answer 5 key questions.

1. Where is the Solaris™ software source?
   In other words is there a CD-ROM drive attached to your system or a remote CD-ROM drive on the network.
2. Is the hardware platform you are installing supported on Solaris™?
   To answer this question, see the Solaris™ Hardware Platform Guide. Also, if your system is on the network, you will need to contact your system administrator to gather the information outlined in Table 2.
3. What Solaris™ components are you going to install?
   Disk space recommendations are listed in Table 3.
4. What language will you be using?
   Solaris™ Multilingual SPARC® supports the following languages: Chinese (simplified and traditional), English, French, German, Italian, Japanese, Korean, Spanish, and Swedish.
5. Is this an upgrade or a new installation?
   During the installation process you will be asked if this is an *initial* or *upgrade* installation.

Once the file copy process is complete you will see the following screen (Fig. 3).

Running WebStart

After the Welcome Screen, click the **Next** button on the graphical display. WebStart will start gathering information to complete the installation.



Welcome

The following screens will gather information about this system.

This information will be used to configure:

- Network
- Name Service
- Date and Time
- Root Password
- Proxy Server Information
- Power Management

To go to the next screen, click Next

Fig. 3. Welcome Screen.

Table 2
Networking Parameters

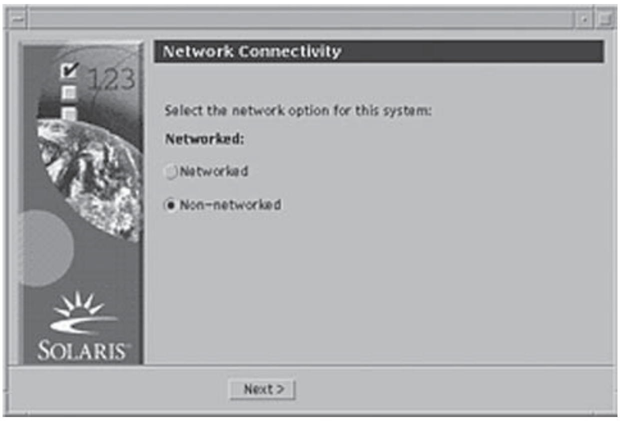
Information	*Example*	*Solaris commands*
Host Name	jordan	uname -u
Host IP address	10.14.121.115	ypmatch system_*name hosts* or nismatch system_*nameHosts.org_dir*
Subnet mask	255.255.255.0	more /etc/netmasks
Type of name service (DNS, NIS, NIS+)	group: files nis passwd: files nis	cat /etc/nsswitch
Domain name	university.edu	domainname
Host name of the name server	brandan	ypwhich
Host IP address of the name server	10.14.110.5	Ypmatch names_*server hosts* or Nismatch name_*server hosts.org_dir*

Table 3
Disk Space Recommendations

Software	*Recommended disk space*
Distribution Plus OEM Support	2.4 Gbytes
Distribution Only	2.3 Gbytes
Developer Support	1.9 Gbytes
End User Support	1.6 Gbytes

The next few steps will configure, the network portion of the installation as shown in Figs. 4–20. Figure 4 begins the process with network connectivity.

If your system will not be connected to a network, then select **Non-networked**. If you select **Networked**, then you will be asked to answer a few more questions. You will need all the information in Table 2. At the screen shown in Fig. 5, you will type in the host name of your system. If you have questions with this step or other steps please see your site system administrator.



Network Connectivity

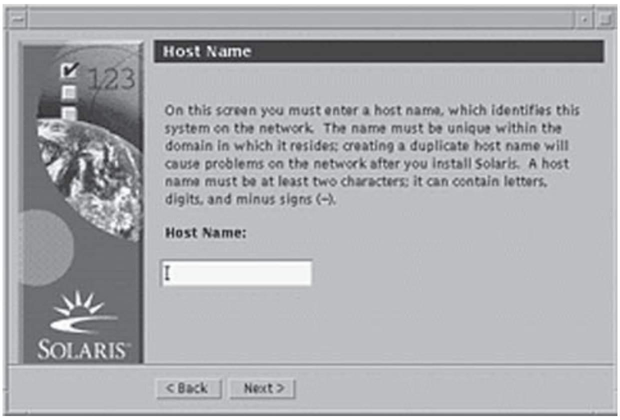
Select the network option for this system:

Networked:

○ Networked

● Non-networked

Fig. 4. Network Connectivity.



Host Name

On this screen you must enter a host name, which identifies this system on the network. The name must be unique within the domain in which it resides; creating a duplicate host name will cause problems on the network after you install Solaris. A host name must be at least two characters; it can contain letters, digits, and minus signs (-).

Host Name:

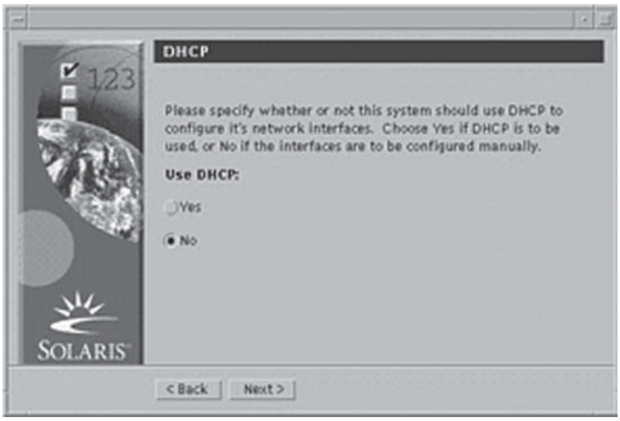
Fig. 5. Host Name.

The next step will configure network addressing. The dialog box (*see* Fig. 6), will ask if you wish to use Dynamic Host Control Protocol (DHCP). The advantage in using DHCP is that it configures the network settings automatically for you. This eliminates potential human error that may occur. If you select No at the DHCP Screen, you will be required to input all necessary network configuration values.

The next screen will require that you input the IP address for your system (*see* Fig. 7). Input the IP address that the system or network administrator has provided for your machine.

The next screen, shown in Fig. 8, specifies your net mask or subnet mask. Input the Netmask or Subnet mask for your system, the default value is 255.255.255.0. In some cases the subnet mask will be different depending upon the sites network configuration. This can be verified with your administrator.

The next screen (*see* Fig. 9), requires a yes/no response to enable the IPv6, the next generation Internet Protocol. It is suggested that you select **No** unless your site is using this new protocol.



DHCP

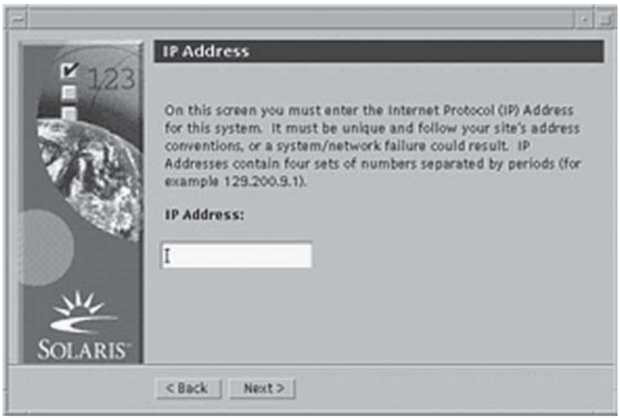
Please specify whether or not this system should use DHCP to configure it's network interfaces. Choose Yes if DHCP is to be used, or No if the interfaces are to be configured manually.

Use DHCP:

○ Yes

● No

Fig. 6. Dynamic Host Control Protocol (DHCP).

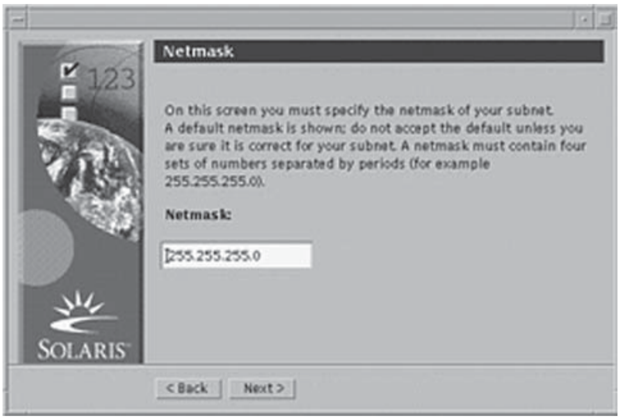


IP Address

On this screen you must enter the Internet Protocol (IP) Address for this system. It must be unique and follow your site's address conventions, or a system/network failure could result. IP Addresses contain four sets of numbers separated by periods (for example 129.200.9.1).

IP Address:

Fig. 7. IP address.



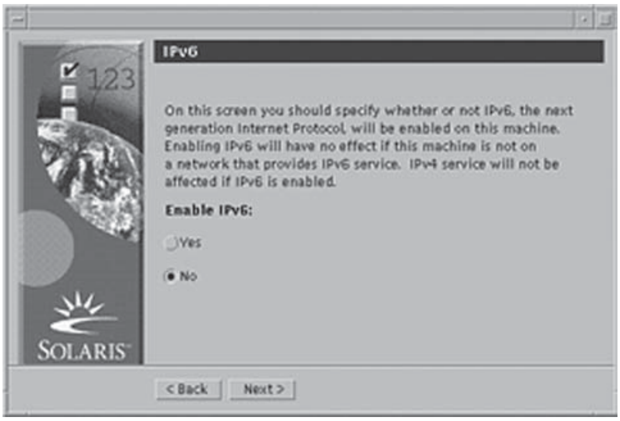
Netmask

On this screen you must specify the netmask of your subnet. A default netmask is shown; do not accept the default unless you are sure it is correct for your subnet. A netmask must contain four sets of numbers separated by periods (for example 255.255.255.0).

Netmask:

255.255.255.0

Fig. 8. Netmask or Subnet Mask.



IPv6

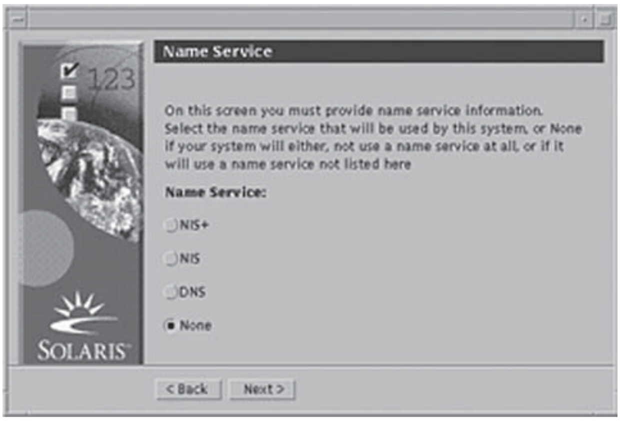
On this screen you should specify whether or not IPv6, the next generation Internet Protocol, will be enabled on this machine. Enabling IPv6 will have no effect if this machine is not on a network that provides IPv6 service. IPv4 service will not be affected if IPv6 is enabled.

Enable IPv6:

○ Yes

● No

Fig. 9. IPv6, the next generation Internet Protocol.



Name Service

On this screen you must provide name service information. Select the name service that will be used by this system, or None if your system will either, not use a name service at all, or if it will use a name service not listed here

Name Service:

○ NIS+

○ NIS

○ DNS

● None

Fig. 10. Name Service.

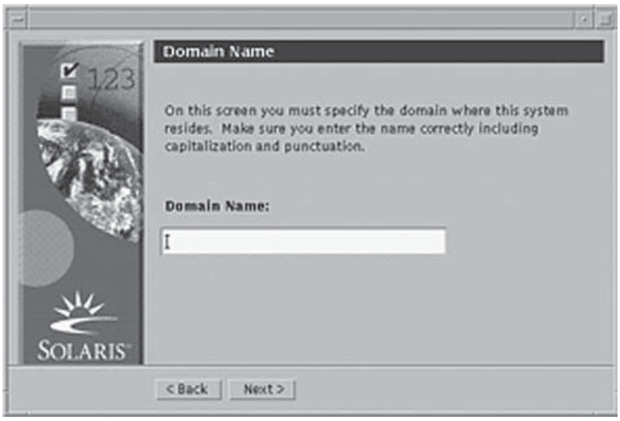
The screen shown in Fig. 10 allows you to select the preferred name service for your system. The name service provides *friendly* name resolution to a physical IP (network) address.

As indicated, the domain name is requested next (*see* Fig. 11). Selecting **DNS**, **NIS**, or **NIS+** will require you to enter a *Domain Name*. An example of a domain name would be "university.edu". The fully qualified name of the machine would be "jordan.university.edu." with "jordan" as the hostname.

If you selected NIS or NIS+ you will be able to specify the name server or it will try to find a name server on your subnet (*see* Fig. 12).

If you selected DNS for the name service you will need to input the IP address of the DNS server. You can input as many as three different DNS server addresses (*see* Fig. 13). Primary and secondary servers are recommended to provide alternative access in case of a DNS server failure. Your network administrator can provide the IP addresses.

If you selected NIS or NIS+ you will need to input the *host name* and the *IP address* of the NIS or NIS+ server (*see* Fig. 14).

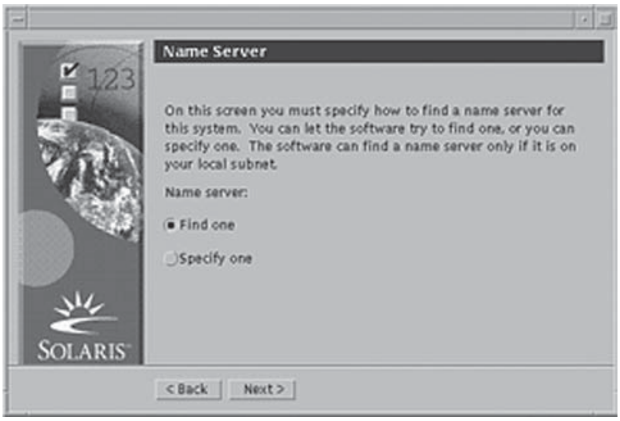


Domain Name

On this screen you must specify the domain where this system resides. Make sure you enter the name correctly including capitalization and punctuation.

Domain Name:

Fig. 11. Domain Name.



Name Server

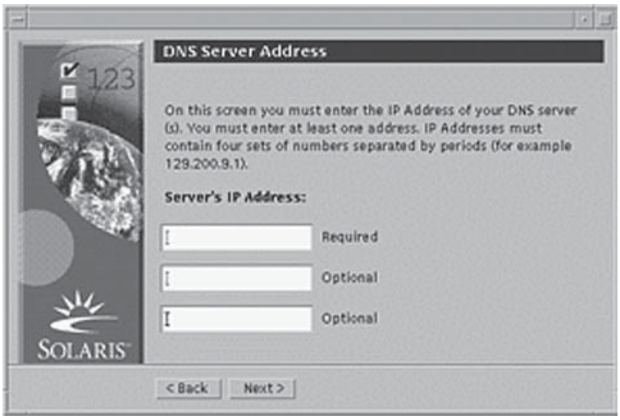
On this screen you must specify how to find a name server for this system. You can let the software try to find one, or you can specify one. The software can find a name server only if it is on your local subnet.

Name server:

● Find one

○ Specify one

Fig. 12. Name Server, Find one.



DNS Server Address

On this screen you must enter the IP Address of your DNS server (s). You must enter at least one address. IP Addresses must contain four sets of numbers separated by periods (for example 129.200.9.1).

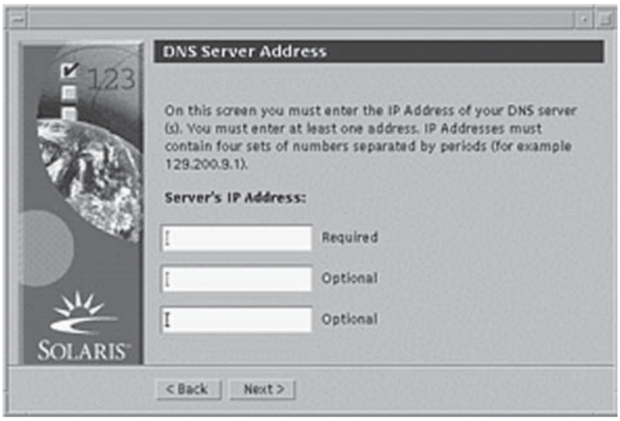
Server's IP Address:

Required

Optional

Optional

Fig. 13. DNS Server Address.



DNS Server Address

On this screen you must enter the IP Address of your DNS server (s). You must enter at least one address. IP Addresses must contain four sets of numbers separated by periods (for example 129.200.9.1).

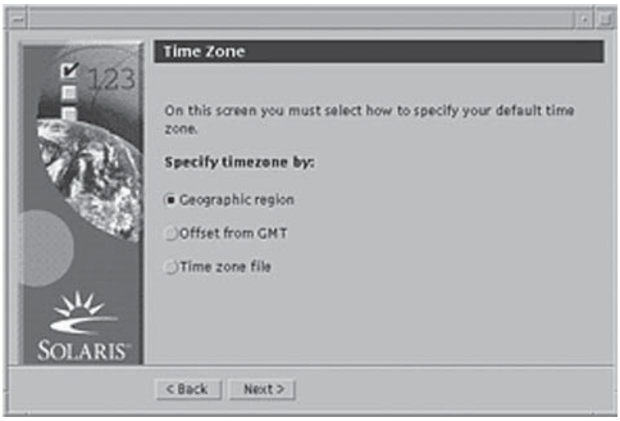
Server's IP Address:

Required

Optional

Optional

Fig. 14. Name Server Information.



Time Zone

On this screen you must select how to specify your default time zone.

Specify timezone by:

● Geographic region

○ Offset from GMT

○ Time zone file

Fig. 15. Time Zone.

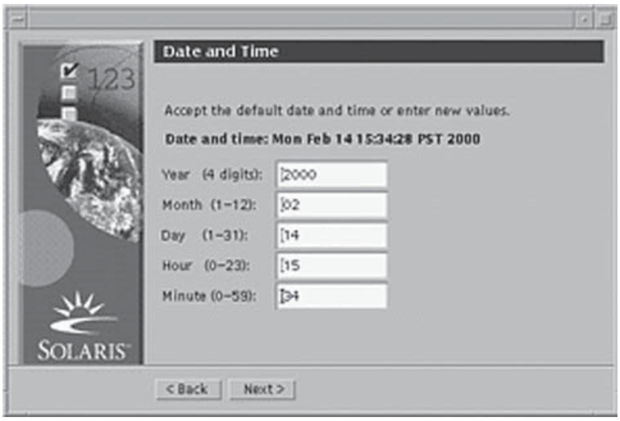
The screen shown in Fig. 15 will allow you to input the geographic region, offset from GMT or a Time Zone File. Time zone files are usually located in the /usr/share/lib/zoneinfo directory.

From the next screen you will need to input the year, month, day, hour, and minutes (*see* Fig. 16). This will set the local time for your system.

The screen in Fig. 17 is where you input the *superuser* password for your system. Please guard this password with the highest level of security. It is the highest level of access to the system. A typical password contains both letters and characters in both upper and lower case. Remember, the Solaris™ (UNIX) operating environment is case sensitive.

Power management software automatically saves the state of the system after it has been idle for 30 min. The software is installed by default and requires the user to either enable or disable the feature (*see* Fig. 18).

Next, there are two selections for connecting to the Internet, direct connection via a gateway or a proxy server. If you select proxy server, you need to provide the host



Date and Time

Accept the default date and time or enter new values.

Date and time: Mon Feb 14 15:34:28 PST 2000

Year (4 digits): 2000

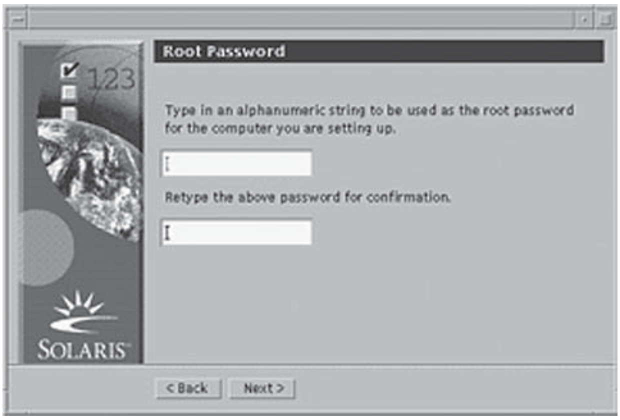
Month (1–12): 02

Day (1–31): 14

Hour (0–23): 15

Minute (0–59): 34

Fig. 16. Date and Time.

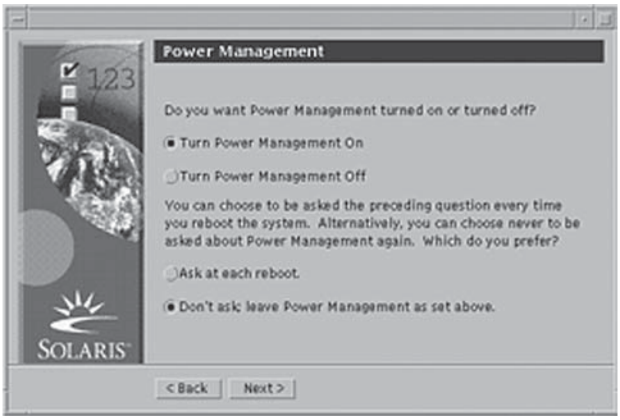


Root Password

Type in an alphanumeric string to be used as the root password for the computer you are setting up.

Retype the above password for confirmation.

Fig. 17. Root Password.



Power Management

Do you want Power Management turned on or turned off?

● Turn Power Management On

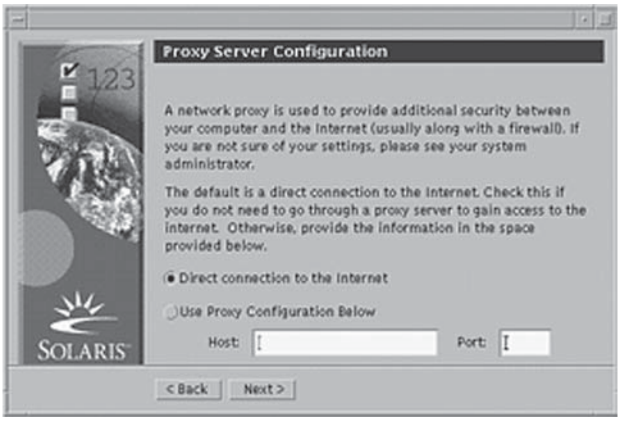
○ Turn Power Management Off

You can choose to be asked the preceding question every time you reboot the system. Alternatively, you can choose never to be asked about Power Management again. Which do you prefer?

○ Ask at each reboot.

● Don't ask; leave Power Management as set above.

Fig. 18. Power Management.



Proxy Server Configuration

A network proxy is used to provide additional security between your computer and the Internet (usually along with a firewall). If you are not sure of your settings, please see your system administrator.

The default is a direct connection to the Internet. Check this if you do not need to go through a proxy server to gain access to the internet. Otherwise, provide the information in the space provided below.

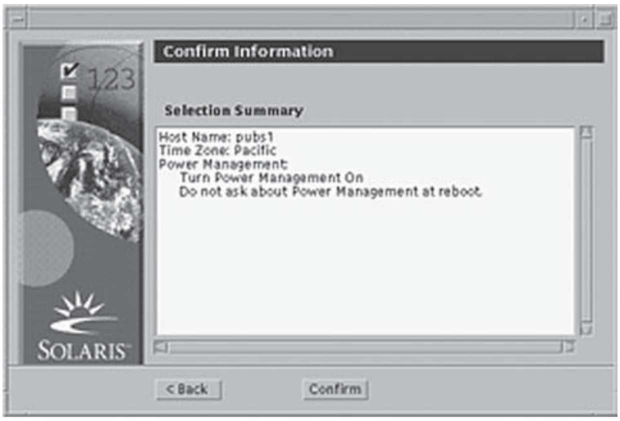
● Direct connection to the Internet

○ Use Proxy Configuration Below

Host:

Port:

Fig. 19. Proxy Server Configuration.

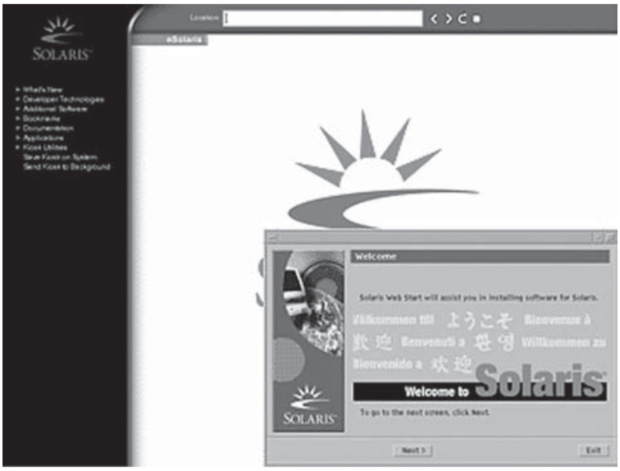


Confirm Information

Selection Summary

```
Host Name: pubs1
Time Zone: Pacific
Power Management:
    Turn Power Management On
    Do not ask about Power Management at reboot.
```

Fig. 20. Confirm Information.



Welcome

Solaris Web Start will assist you in installing software for Solaris.

Välkommen till ようこそ Bienvenue à 歓迎 Benvenuti a 환영 Willkommen zu Bienvenido a 欢迎 Welcome to Solaris

To go to the next screen, click Next.

Fig. 21. Main Menu.

name and the port your proxy server is on (*see* Fig. 19). Your local network administrator can advise.

The screen shown in Fig. 20 allows you to verify the settings before they are committed to the system. When you click on the **Confirm** button, the system will write the configuration information to the system. The **back** button permits scrolling to previous screens, to change configuration parameters.

Once you have confirmed your settings, you have completed the operating system installation (*see* Fig. 21). You are now ready to work in the Solaris™ Operating Environment.

Glossary and Abbreviations

Application Is a program designed to perform a specific function directly for the user or, in some cases, for another application program. (Examples of applications include word processors, database programs, Web browsers, development tools, drawing, paint, image editing programs, and communication programs).

CD-ROM (Compact Disc, read-only-memory) is an adaptation of the CD that is designed to store computer data in the form of text and graphics, as well as hi-fi stereo sound.

Command Line Interface (CLI) Is a user interface to a computer's operating system or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command, and so forth.

Default Router Is a device or, in some cases, software in a computer, that determines the next network point to which a packet should be forwarded toward its destination. The router is connected to at least two networks and decides which way to send each information packet based on its current understanding of the state of the networks it is connected to.

DHCP Dynamic Host Configuration Protocol. DHCP is a communications protocol that lets network administrators manage centrally and automate the assignment of Internet Protocol (IP) addresses in an organization's network. Without DHCP, the IP address must be entered manually at each computer and, if computers move to another location in another part of the network, a new IP address must be entered.

DNS The Domain Name System. DNS is the way that Internet domain names are located and translated into Internet Protocol addresses. A domain name is a meaningful and easy-to-remember "handle" for an Internet address.

Domain Name A domain name locates an organization or other entity on the Internet. For example, the domain name www.cnn.com locates an Internet address for "cnn.com" at Internet address: 64.236.24.20.

Dual Boot A dual boot system is a computer system in which two operating systems are installed on the same hard drive, allowing either operating system to be loaded and given control. When you turn the computer on, a boot manager program displays a menu, allowing you to choose the operating system you wish to use. A boot manager works by replacing the original Master Boot Record (MBR) with its own so that the boot manager program loads instead of an operating system.

EISA Is a standard bus (computer interconnection) architecture that extends the ISA standard to a 32-bit interface. It was developed in part as an open alternative to the proprietary Micro Channel Architecture (MCA) that IBM introduced in its PS/2 computers. EISA data transfer can reach a peak of 33 megabytes per second.

Giga Byte (GB) A gigabyte is a measure of computer data storage capacity and is "roughly" a billion bytes.

GMT Is the standard time common to every place in the world. Formerly and still widely called Greenwich Mean Time (GMT) and also World Time, This nominally reflects the mean solar time along the Earth's prime meridian. (The prime meridian is 0° longitude in the 360 lines of longitude on Earth.

Graphics Monitor A monitor that is capable of displaying graphics.

Hostname Any computer that has full two-way access to other computers on the Internet. A host has a specific "local or host number" that, together with the network number, forms its unique IP address. In this context, a "host" is a node in a network.

IP Internet protocol. The Internet Protocol (IP) is the method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet. When you send or receive data (for example, an e-mail note or a Web page), the message gets divided into little chunks called packets. Each of these packets contains both the sender's Internet address and the receiver's address.

IP Address An IP address has two parts: the identifier of a particular network on the network and an identifier of the particular device (which can be a server or a workstation) within that network.

ISA Industry standard architecture. ISA is a standard bus (computer interconnection) architecture that is associated with the IBM AT motherboard. It allows 16 bits at a time to flow between the motherboard circuitry and an expansion slot card **IPv6** IPv6 (Internet Protocol Version 6) is the latest level of the Internet Protocol (IP) and is now included as part of IP support in many products including the major computer operating systems. IPv6 has also been called "IPng" (IP Next Generation). Network hosts and intermediate nodes with either IPv4 or IPv6 can handle packets formatted for either level of the Internet Protocol. The most obvious improvement in IPv6 over the IPv4 is that IP addresses are lengthened from 32 bits to 128 bits. This extension anticipates considerable future growth of the Internet and provides relief for what was perceived as an impending shortage of network addresses.

JAVA Java is a programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model.

Name Server A server much like a DNS Server that translates Host Names to IP Address or IP Address to Host Names.

NIS Network Information System. NIS is a network naming and administration system for smaller networks that was developed by Sun Microsystems. NIS+ is a later version that provides additional security and other facilities. Using NIS, each host client or server computer in the system has knowledge about the entire system. A user at any host can get access to files or applications on any host in the network with a single user identification and password.

OEM An OEM (original equipment manufacturer) is a company that uses product components from one or more other companies to build a product that it sells under its own company name and brand. IBM is an example of a supplier to the OEM market (and IBM is also an OEM itself since it uses other companies' parts in some of its products).

PCI Peripheral component interconnect. PCI is an interconnection system between a microprocessor and attached devices in which expansion slots are spaced closely for high-speed operation.

Proxy Server A proxy server is a server that acts as an intermediary between a workstation user and the Internet so that the enterprise can ensure security, administrative control, and caching service. A proxy server is associated with or part of a gateway server that separates the enterprise network from the outside network and a firewall server that protects the enterprise network from outside intrusion.

SPARC® Scalable processor architecture. SPARC® is a 32- and 64-bit microprocessor architecture from Sun Microsystems that is based on reduced instruction set computing (RISC). SPARC® has become a widely used architecture for hardware used with UNIX-based operating systems, including Sun's own Solaris systems.

Solaris™ Solaris™ is the computer operating system that Sun Microsystems provides for its family of Scalable Processor Architecture-based processors as well as for -based processors.

Subnet Mask is an identifiably separate part of an organization's network. Typically, a subnet may represent all the machines at one geographic location, in one building, or on the same local area network (LAN).

Superuser A user that has administrative privileges on the system. The superuser account can perform any function on the system.

UNIX is an operating system that originated at Bell Labs in 1969 as an interactive time-sharing system. Ken Thompson and Dennis Ritchie are considered the inventors of UNIX.

x86 Is a generic name for the series of Intel® microprocessor families that began with the 80286 microprocessor. This series has been the provider of computing for personal computers since the 80286 was introduced in 1982. x86 microprocessors include the 386DX/SX/SL family, the 486DX/SX//DX2/SL/DX4 family, and the Pentium 3 family.

Suggested Readings

Watters, P. A. (2001) Solaris Administration: A Beginner's Guide, Osborne McGraw Hill, Berkeley, CA.

Kasper, P. A. and McCellan, A. (1995) Automating Solaris Installations: A Custom Jumpstart Guide with Disk, Prentice Hall, Upper Saddle River, NJ.

Wong, B. L. (1997) Configuration and Capacity Planning for Solaris Servers, Prentice Hall, Upper Saddle River, NJ.

Ledesma, R. (1994) PC Hardware Configuration Guide for DOS and Solaris, Prentice Hall, Upper Saddle River, NJ.

15 Sun System Administration

Bryon Campbell

Introduction

This chapter will describe the basic day-to-day administrative tasks necessary to manage a single or multi-user system. The primary source of information has been gathered from the Sun documentation website and the experience of the author (*see* Website: http://docs.sun.com). Additional and detailed information is always available as part of your Solaris™ User and System Administration Guides or local UNIX/Solaris™ experts.

User Administration

Managing User and Group Account Databases

One of the main tasks for the system administrator is to manage user access to the Solaris™ system(s). This includes the creation of user accounts, password changes, and user security. The file that contains this information is located in the directory: **/etc/passwd**.

The file format is text and utilizes single lines of information to define each user. The format for the information is: **username:x:uid:gid:commment:home_directory:login_shell**. Table 1 describes the field names for the **passwd** file.

In many cases there are system parameters that predefine the minimum and maximum values for user accounts and passwords.

The *User Account* is defined by the username that uniquely identifies the user who is logging into the system and accessing resources. The author recommends that the system administrator defines a standard for usernames. This will reduce the potential for problems as the user base grows.

The *Password* is the parameter that authenticates the user to permit access to the system. Again the author recommends defining a standard for passwords (e.g., minimum length and combination of characters and numbers). The location for the encrypted password file is in: **/etc/shadow**. This file is only accessible by the root user (highest level of access).

The format of the **/etc/shadow** file is much like that of the **/etc/passwd** file: **username:password:lastchg:min:max:warn:inactive:exprire:flag**. Table 2 describes the various fields.

Table 1
/etc/passwd File Fields

Field Name	*Description*
Username	Unique username
X	Locator for users encrypted password
Uid	**u**ser **id**entification number—used to define user system security
Gid	**g**roup **id**entification number—used to define the security group to which the user belongs
Comment	Holds additional information about the user or user account
home_dir	Directory that the user has been assigned for storing resources (e.g., files or programs)
login_shell	Defines the users default login shell (e.g., sh, csh, ksh)

Table 2
/etc/shadow File Fields

Field Name	*Description*
Username	Unique username
Password	13 character encrypted string—users password
Lastchg	Date password was last changed
Min	Minimum days required between password changes
Max	Maximum days the password is valid
Warn	Number of days the user is warned of password expiration
Inactive	Number of days the account can be inactive (no login). If left blank the account will always be active
Expire	Expiration of the account should be set for temporary or contract workers
Flag	Not in use

The *User ID* uniquely identifies each user on the system. This is analogous to the name, associated with the user's account. The range for UIDs is quite large, 0 to approx 2.1 billion. Solaris™ reserves ranges of IDs for administrative purposes as shown in Table 3.

The *Group ID*s are similar to the UID but it identifies a group of users. This allows a system administrator to apply security to resources to which a group of users have access. It minimizes the amount of administration that would be required to apply security privileges to each individual user (UID).

The GID information is stored in the **/etc/group** file as shown in Table 4. It emulates the format used in the **/etc/passwd** and **/etc/shadow** files: **groupname: password:gid:user_list**.

Table 3
Reserved Solaris™ UIDs

UID	*Description*
0	Root or *superuser*
1–99	Daemons and system maintenance processes
60001	Special user access *nobody*, used in specific maintenance activities
60002	Special user access *noaccess*, used in the same way as *nobody*
65534	Special user access *nobody4*, used for backward Solaris operating system compatibility.

Table 4
GID Parameters

Field Name	*Description*
Groupname	Symbolic name of the group (maximum of 8 characters)
Password	Not used in Solaris 8, carry over from previous Solaris operating systems
Gid	Unique numerical number
User_list	Comma separated list of users

Table 5
useradd or **usermod** Parameters

Option	*Parameter Name*	*Description*
-c	comment	User comment
-d	pathname	Path for user home directory
-g	groupname or GID	Default groupname or group ID for the user
-e	expire	Expiration for the user account
-f	inactive	Maximum inactivity period for a user account
-u	UID	UID to use for the user
-s	shell	Default login shell
-l	newname	Used to change user name

Managing User and Group Accounts

There are several commands that are used to manage user accounts. For illustrative purposes the use of a few commands will be demonstrated. All options are case-sensitive.

The **useradd "username"** command is executed by the root user (e.g., useradd christina). This command would create a user named "christina". There are several parameters that can be passed in the command line as shown in Table 5. These could also be used to change user parameters with the **usermod** command.

Table 6
passwd Parameters

Option	*Parameter Name*	*Description*
-f	username	Force the user to change password at next logon
-x	days username	Number of days the password is valid
-w	days username	Number of days before the password expires
-n	min username	Number of days a user has to wait to change their password
-l	username	Locks the specified user account

The **usermod "option" "username"** command permits the *root user* to change the characteristics associated with a specific user account. The parameters listed in Table 5 can be applied to **usermod** command.

The **userdel "username"** command is used to delete a specific user account. This command deletes the account from the system but does not remove any resources associated with the account.

The **passwd "option" "username"** command is used to change a users password. The **passwd** command can be executed by the account owner or the root user. An example of the parameters that can be used with the **passwd** command are shown in Table 6.

The **groupadd -g "gid" "groupname"** is similar to the **useradd** command. This command will create a group when provided a group ID (GID) and groupname.

The **groupmod -n "newname" "currentgroupname"** command will change the name of a current group.

The **groupdel "groupname"** deletes a group from the system.

System Process Administration

System processes are also known as *daemons*. Daemons are processes that allow the Solaris™ Operating Environment to be single and multi-user systems. They are the traffic cops of the operating system. These processes make computer resources available, manage interactions with other processes, in order to keep them from crashing into one another. They are the backbone of any operating system.

System Startup and Shutdown

There are several modes of operation when the Solaris system starts up. The *init* daemon is crucial to this process as *init* controls the initialization of the system. The initialization process and parameters are settable from the **/etc/inittab** file. The main function of *init* is to spawn the necessary daemon processes from the configuration file **/etc/inittab**.

Information stored in the **inittab** file handles many situations that arise during the boot process, (*see* Table 7).

Shutdown has been made easy with the **shutdown** command. The parameters for the shutdown command are: **#shutdown -i "run_level" -g "grace_period" -y**, where **-i** specifies the run level for the shutdown (*see* Table 8), **-g** specifies the period of time before the shutdown starts, and **-y** supplies the confirmation parameter for shutdown. If this is omitted the system will ask if you want to really shutdown.

Table 7
init Actions

Action	*Description*
boot	Initialization of a full reboot
bootwait	Transition from single-user to multi-user system
initdefault	Default run level (on all systems)
off	Ensures process termination
ondemand	Similar to re-spawn
powerfail	State when a power failure is detected
powerwait	Waits for process to terminate
respawn	Ensures specified process are running
sysinit	Safeguard to allow administrator to specify run level
wait	Starts a process and waits until it's started

Table 8
Solaris Run Levels

Run Level	*Init State*	*Type*	*Description*
0	Power Down	Power-down	Shutdown system to safely power off system
1	Administrative	Single-user	Access to user files with user logins
2	Multi-user	Multi-user	Normal operations, multiple users except the NFS daemon does not start
3	Multi-user with NFS	Multi-user	Normal operation with NFS daemon started
4	Alternative Multi-user	Multi-user	Currently not in use
5	Power Down	Power-down	Shutdown system to safely power off system with automatic power off if available
6	Reboot	Reboot	Shutdown system and reboot to multi-user state
s or S	Single-user	Single-user	Start as single user system with all resources available

Startup and Shutdown Run Levels

The systems *run level* or *init state* defines what services will be made available during startup or shutdown. Solaris™ has eight run levels defined in the **/etc/inittab** file. Table 8 describes the eight available run levels.

System Process Management

System processes are applications that run in support of the operating environment. These processes manage or control files, security, printing, network, and user applications. Solaris™ is a UNIX-based operating system and supports a multi-user

Table 9
ps Parameters

Parameter Name	*Description*
-a	Frequently requested processes
-e or -d	All processes
-c	Processes in scheduler format
-f or -l	Complete process information
-g or -G	Group information
-j	SID and PGID
-p	Processes for a specific process
-s	Session leaders
-t	Specific terminal process information
-u	Specific user process information

environment. These system processes ensure that everyone who is accessing the system is provided a secure portion of the environment enabling them to perform the tasks they are trying to accomplish.

There are many tools available in Solaris™ to assist system administrators with the management of the system. In the UNIX environment users have the ability to *spawn*, i.e., *create* new process. This sometimes creates problems and system administrators need tools that enable them to monitor or delete processes that are depleting system resources.

The most useful command is the **ps** command. There are many "case sensitive" parameters that can be passed with the **ps** command that will allow the system administrator to analyze the current state of the system. Table 9 shows a list of the parameters and what they provide.

Signals can also be sent to specific process that will stop or halt errant processes using the **kill** command. There are two parameters that are used to affect running processes. All signals are defined in the **<signal.h>** file. In **kill "signal_type" "pid"**, a **#kill -9 235** command would kill process 235, whereas a **#kill -3 236** command would kill process 236. Note: **-9** terminates any process; **-3** exits a process gracefully.

Network Basics

A network is the glue that connects systems together. The following basic networking concepts and commands are outlined.

Network Layers

There are several communication layers that achieve the connectivity that is needed to share information between one or more systems. Network protocols are the basic components that constitute the various network layers. These protocols are used to ensure that information is delivered via the network in a timely and accurately manner (*see* Table 10).

Table 10
Network Layers

Network Layers	Layer Function
Application Presentation Session	Application data
Transport Network	Host to host communications
Data Link Physical	Network access

As outlined in Table 10, each layer has a specific function as follows:

- Application, is where applications reside. It is the end user interface to the operating system (UNIX, LINUX, etc).
- Presentation, isolates the lower layers from the application data format. It converts data from the application into a common interchangeable format.
- Session, organizes and synchronizes the exchange of data between application processes. The session layer can be thought of as a timing and flow control layer.
- Transport, provides the transfer of data from a source end open system to a destination end open system. It establishes, maintains, and terminates communications between two machines while maintaining data integrity.
- Network, provides physical routing of the data, determining the path between the systems. The network layer handles all routing issues. The routing function will evaluate the network topology to determine the best route to send data.
- Data Link, monitors the signal interference on the physical transmission media. It also detects and in some cases corrects errors that occur, and provides the means to control the physical layer.
- Physical, is the lowest layer of the 7-layer OSI model representing the media and electrical components required for the transmission of data.

TCP/IP

The Transmission Control Protocol/Internet Protocol (TCP/IP) is the backbone of most networks today. It has become the standard for Internet-based network communications. This network protocol enables heterogeneous systems to communicate with one another. The Transmission Control Protocol (TCP) is a communications protocol that provides reliable exchange of data. It is responsible for assembling data passed from higher-layer applications into standard packets to ensure that the data is correctly transferred. Whereas the Internet Protocol (IP) moves the packets of data assembled by TCP across networks. It uses a set of unique addresses for every device on the network to determine its destinations and routing.

Network Addressing

The IP address is a 32-bit address that is divided into two sections: the prefix for network address and the suffix for host address. This is designed to make IP packet routing extremely efficient. IP addresses are presented in dotted decimal notation. The dotted decimal address range is from 0.0.0.0 through 255.255.255.255. IP Addresses are grouped in classes, as shown in Table 11.

With the expansion of the Internet there are currently only class 'C' addresses available and the standards committee is considering various means to extend the address space to meet the growing demand.

Network Services

There are other components of the TCP/IP Protocol that enable network administrators and users to monitor, troubleshoot, and easily share information. These tools are Kerberos (security), Simple Network Management Protocol (network management), Network File Server (UNIX File Systems), Remote Procedure Calls (programming tool), Trivial File Transfer Protocol, User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP).

There are several ports available to the TCP protocol that can be used for specific functions. The daemon processes on the remote system will monitor or listen to these ports for connection requests. A detailed list of the ports are available in the file /etc/ services. Some of the more common ports are listed in Table 12.

Network Commands

There are several commands that allow users or administrators to manage network connections. A few of the common commands are described in Table 13. Additional Information is available in the user manuals or the "man" pages within the Solaris™ operating system.

FTP Administration

File Transfer Protocol (FTP) is a mechanism for transferring files between hosts. This common Internet protocol has been used for many years. FTP is a TCP/IP protocol that is referenced in RFC 959. This is one way to allow the sharing of publicly available files. Little or no security is required to access this information. There are various levels of security that can be incorporated to protect the information.

FTP Commands

FTP is a command rich environment that has made it extremely easy for users to upload and download files. Some of the FTP commands that are available are shown in Table 14.

FTP Access

The most common user of FTP is the *anonymous* account. This is typically a publicly accessible account and grants users access to public information.

FTP Interface

The interface can be command line or Graphical User Interface (GUI). The GUI interface is the most intuitive interface from a user's perspective. The command line

Table 11
IP Address Classes and Values

IP class	*Prefix bits*	*Number of networks*	*Suffix bits*	*Number of hosts*
A	7	126	24	16777214
B	714	16384	16	65532
C	21	2097152	8	254

Table 12
TCP Port Numbers

Port Number	*Service Name*	*Description*
7	echo	Echo's back from a remote host
21	ftp	Used to control the File Transfer Protocol (FTP)
22	ssh	Encrypted communications with remote systems
23	telnet	Terminal access to remote systems
25	smtp	Mail transfer protocol
514	shell	Executing programs on remote systems
80	http	Web server

Table 13
Network Commands

Tool	*Description*
arp	Address Resolution Protocol (ARP) is a cache table that is updated based on the presence of MAC address (network card physical address) in broadcast traffic.
ping	Tests the reach ability of remote systems. It uses ICMP, an echo request to test round trip connectivity.
snoop	You must run this as the root user. Snoop will monitor all network traffic passing by the system it is running on. The interface will be running in promiscuous mode to monitor all network traffic. You should contact your network administrator before you run snoop to ensure you will not be violating privacy policies.

interface can be difficult unless one is familiar with FTP. You can find freeware FTP products that will run on Solaris™ operating system on the Internet (*see* Website: http://www.sunfreeware.com).

Device Administration

One of the most important aspects of managing a system is managing the resources connected to the system. These resources are typically storage devices but other devices such as scanners, video, and audio devices can be attached to a system. Management of these devices is relatively simple in the Solaris™ environment. Solaris™

Table 14
FTP Command Options

FTP Command	*Description*
binary	Set file transfer type to binary
cd	Change remote working directory
close	Terminate current FTP session (don't exit from client)
delete	Delete specified file
debug	Set debugging mode to on
dir	List contents of remote directory
disconnect	Terminate current FTP session and exit client
form	Set file transfer type to binary or ASCII
get	Download specified remote file
help	Help with command usage
ls	List contents of remote directory
mget	Get multiple files from remote system
mkdir	Make directory on remote system
mode	Set transfer mode to binary or ASCII
open	Connect to a remote system
put	Upload a file to a remote system
pwd	Print the current working directory
quit	Terminate FTP session and exit
rename	Rename specified file
rmdir	Remove directory on remote system
send	Upload a file to the remote system
status	Show the current status of the FTP session
type	Set file type for file being transferred.
verbose	Echo detail information to screen
?	Print help on local system

supports dynamic configurations of devices on some of the SPARC® platforms. This makes it simple for administrators and users to manage large and small systems. Information for each device is stored in a *device file*, which is in the /dev directory (*see* Table 15).

Storage Devices

There are several storage devices supported in the Solaris™ operating system. CD-ROMs, Zip Drives, Jaz Drives, Tape Drives, Hard Drives, and Floppy Disks are just a few of the device types that are supported. Hard drives are the most common device. In some configurations it may be perferred to configure your hard disks with Redundant Array of Independent Disks (RAID) support. A RAID configuration will provide you an extra level of redundancy that will protect the data in the case of a

Table 15
/dev Directory

/dev	*Description*
/dev/console	Console device
/dev/null	Discarded output
/dev/hme	Network interface device
/dev/ttyn	Terminal devices attached to the system
/dev/dsk	Files for disk partitions

Table 16
Disk Partition

Partition Number	*Directory*
0	/ - Root level directory
1	Server *swap* space
2	Whole disk
3	/export
4	Client *swap* space
5	/opt
6	/usr
7	/export/home

hard disk failure. The most common RAID configurations are RAID-0 (disk mirroring) and RAID-5 (disk stripping with parity bit; *see* Glossary).

When a hard disk volume is created and configured the Solaris™ operating system will create several partitions. The partitions that are created on a SPARC® system are listed in Table 16.

List System Devices

To check for the type of devices connected to your system you can issue the **prtconf** command. If you see devices that are not connected, don't be concerned because Solaris loads device drivers based on the usage of the device. This allows the system to optimize resources, ensuring optimal performance.

Adding New Devices

Prior to adding a device to your system, there are several steps to be follow.

1. Purchase a device that is supported by the Solaris™ operating system.
2. Check your system to ensure you have physical capacity for the new device (internal and external).
3. Check your SCSI bus to ensure you have enough device locations to add the new device.
4. Follow the directions for installation provided by the vendor you purchased the device from.

Table 17
Physical Disk

Disk Area	*Description*
Platter	Magnetic disk that turns on a shaft.
Cylinder	A cylinder refers to the location of all the drive read and write heads, typically accessing multiple tracks.
Track	This refers to one of the concentric circles of data on disk media.
Block	A specific area within a track that data is written to.

File System Administration

Managing the file system will allow you to control the resource usage for any given system. You first need to understand how the file system is structured. The Solaris™ UNIX File System (UFS) came from the Berkley Software Design, Inc. (BSD) environment and allows single or multiple users to utilize the Solaris™ operating system. The following briefly describes how to utilize the partitions of the disk to store and retrieve information.

Basic Disk Structure

The disk is made up of four basic components as shown in Table 17. The typical disk geometry is composed of platters that are magnetically coated. This coating allows the system to write information to a very small area of the disk using 1's and 0's, binary notation (1 = on, 0 = off). There are typically several platters in a disk drive unit.

The Solaris™ UFS is segmented into specialized areas (*see* Table 18). This differs from the original format and creates a higher level of reliability than before.

Formatting A Disk Device

There are several steps that take place when a file system is created for the first time. How the disk space will be used will dictate how the file system should be configured. Typically larger partitions for data volumes are created to adjust capacity as users store data on the system.

When executing the format command Solaris™ will display the available devices on the system. If there is more than one disk, you will be prompted to enter the device to format. After selecting the device the system will display a menu that provides several options as shown in Table 19.

Note: If for some reason you select a device you don't want to format, you can stop the program by issuing the **ctrl-c** command.

Creating A File System

Upon formatting the new device, a new file system must be created. The command for creating a new file system is **#newfs devicename**. The device name is the logical device pathname on the device where the new files system will be created. An example is **#newfs /dev/dsk/c2t3d1s5**. (*See* Glossary for a description of **c2t3d1s5**.)

Table 18
UFS Disk Segments

Disk Segment	*Description*
Superblock	Contains information on disk geometry, free blocks, and inodes. Multiple copies are made to ensure reliability.
Inode	One inode per file. Ownership, time stamp, size and links are kept in this file.
Data Block	Data for the file being stored. The size of the block depends on how the disk was initialized (e.g., 512K block).

Table 19
Format Menu Commands

Menu Item	*Description*
Disk	Select the disk you want to format
Type	Define the type of disk your are formatting
Partition	Define the partition table
Current	Describe the current disk (disk name)
Format	Format and analyze the disk
Repair	Repair defective sectors
Label	Provide the label name for the disk
Analyze	Do a surface analysis of the disk
Defect	Defect management list (blocks)
Backup	Search for backup labels
Verify	Read and display labels
Save	Save the new disk partition definitions
Inquiry	Show vendor product and revision
Volname	Define eight character volume name
Quit	Leave format menu

Various switch options can be used with the **#newfs** command that allow you to manipulate the disk configuration after the file system is created.

The -m parameter permits one to adjust the amount of reserved space on the file system. The -o parameter permits one to specify whether the system should optimize the disk for space or allocation time. Depending upon your usage of the newly formatted device you may want to consider these options.

Having formatted the storage media the disk must be made either available or unavailable to the operating system. A device must be mounted as part of the system directory tree. There are many options for the **mount** command that may be considered as shown in Table 20. The two commands that assist with this operation are **mount** and **unmount**. The normal syntax for making the resource available is **#mount device directory**. An example is **#mount /dev/dsk/c2t3d1s5 /etc**.

Table 20
Mount Options

Mount Option	*Description*
rw	Mount device with read-write capability
ro	Mount device with read-only capability
nosid	Eliminates use of user or group identifier bits
remount	Remount an already mounted device

This command would mount the formatted disk in the /etc directory. The directory that mounts the device must exist. It should be noted that this superimposes the directory specified with the new system device.

To list the devices mounted on the current system, the mount command from the prompt is issued and the mounted devices on the system will be displayed. By issuing the **mount -v** (verbose) parameter with the mount command, the system will display detailed information on all mounted devices. In contrast, the unmount command makes devices or directories unavailable to the system. The **#unmount directoryname** command makes directory unavailable to the system and the **#unmount devicename** makes the device unavailable to the local system.

In Solaris™ the volume manager daemon manages access to CD-ROM and removable media. The *vold* daemon listens for insertion of media in these removable media devices. When the daemon detects that a CD-ROM has been inserted into the drive it invokes the **rmmount** command to mount the removable media. The vold daemon is also responsible for ejecting removable media for the user. If the vold daemon has frozen, the **eject** command can be used to eject removable media (CD-ROM, Floppy Disk, ZIP Drive).

There are two important maintenance activities that will enable the system administrator to manage storage resources more efficiently. The **#df** command displays device capacity information. This command provides:

- Filesystem: Name of the device or directory
- Kbytes: Total capacity of the device
- Used: Space used on the device
- Avail: Amount of space available on the device
- Capacity: Percentage of device space being used
- Mounted on: Directory mount point for the device

There are several optional parameters that can be passed with the **#df** command.

As a system administrator you also have the ability to define and set quotas on devices. This helps manage the utilization of storage space and prevents any one user from filling a volume and potentially disabling the system. Quotas can be set user by user for any given volume. To set quotas on a volume for a user, follow steps 1–7.

1. Find the etc/vfstab file as the root user.
2. Edit the etc/vfstab file and add the “rq” flag to the mount option field for the directory you want to establish quotas on (e.g., /usr/jordan).
3. Change to the directory you want quotas to be established (e.g., /usr/jordan).

4. Set the permissions on the /usr/jordan to read-write for *root* user only.
5. To set quotas on the directory, use the **edquota** command and set the *hard* and *soft* limits for storage space. The hard parameter sets the maximum limit. The soft parameter sets the overquota or temporary space for a user. An example is **#fs /usr/jordan blocks (soft=2500, hard=5000)**.
6. You can check to see if quotas are set by issuing the **quota** command.
7. To enable quotas once they have been defined, you can issue the **quotaon** command from *root user* ID.

Available storage space is something that you as an administrator need to be aware of and utilizing quotas is the easiest way to ensure you don't see problems with runaway storage from users or errant processes.

Backup Administration

System backup is critical to the stability/recoverability of the system. It allows restoration of files either deleted accidentally (user initiated) or corrupted. When defining the backup strategy consider disaster recovery if the system mission is critical to your operations. Disaster recovery plans are detailed in nature and should be tested at least once a year. This can also be a costly endeavor so be sure you can justify the additional cost associated with disaster recovery.

Backup Media

There are several ways to approach the selection of backup media. The amount/size of data you backup will dictate the best approach for your environment. There are 3 types of media you can direct your backup to:

1. Tape: Reliable and reasonable capacity (DLT: 40GB–80GB).
2. Optical: Could be used for near-line storage (CD-ROM and DVD).
3. Disk: More expensive but fast and readily available (RAID configuration ensures redundancy).

Backup Methodology

There are two methodologies utilized to backup systems. In *Full Backup*, a snapshot image of all the files on the system is created. This is typically executed on off-hours (evening) when the system is not being utilized. Alternatively, the *Incremental Backup* only effects those files that have changes since the last full backup. This is a time-saving and efficient process. An example schedule is a full backup on Saturday starting at 11:00 pm, and an incremental backup every day but Saturday, starting at 11:00 pm.

Note: If you have a database system running, you will need to consult your user/administrator guide for backup of the database. In many cases the database process will have to be stopped to ensure a complete backup. If the database file is open, the backup program may not be able to accurately backup the opened file. This is true for any file that maybe used by processes running on the system.

Backup Process

There are a few means to execute a backup of the system. The easiest way is with the **tar** command. **tar** is used to create and extract files from an archive file. However,

tar does not compress the file like other commercially available products. Compression can be achieved using the Solaris™ **compress** command.

Another method is using the Solaris™ **ufsdump** and **ufsrestore** utilities. These utilities only work on UNIX (UFS) file systems.

ufsdump allows you to create your full and incremental backup schedules. **ufsrestore** is easy to use and works much like the **ufsdump** command. You can utilize this utility in interactive mode by passing the "i" parameter with the **ufsrestore** command.

Commercially available backup products tend to be more feature rich than **ufsdump** and **ufsrestore**. For the Solaris™ Operating Environment there are two excellent choices, Legato® Networker (*see* Website: http://www.legato.com) and Veritas® Netbackup (*see* Website: http://www.veritas.com).

Each of these products have their benefits, so you and your system administrator need to decide if a commercially available product is the best choice for your system.

Web Server Administration

One of the most popular technologies today is web-based technology. This crosses all boundaries of information today, from research, business, consumer, and entertainment. The backbone protocol for the web is HyperText Transfer Protocol (HTTP), which uses Universial Resource Locators (URLs) embedded in HyperText Markup Language (HTML) files that display information in a web browser interface (Netscape Navigator® and Microsoft Explorer®). The combination of these components allows people to surf the World Wide Web (WWW) for anything you can imagine. The Solaris™ environment Apache Server is the component that will make your server WWW enabled.

Apache Web Server

The Apache Web Server product is free from the Apache website where the necessary components can be downloaded (*see* Website: http://www.apache.org). The downloaded file is in *tar* or *tar.Z* (compressed) format. Utilize the released version unless there are specific features in the beta version you need. You should also download or print the release notes specific to the product you have downloaded. These documents contain useful information such as vendor-specific bug fixes, installation procedures, and major application changes.

Installing Apache Web Server

Before installing the software on your system, the following pieces of information are required:

- Your IP Web Server Address (IP address for the Internet).
- Add your server to DNS (Name resolution for your web server).

Steps for Configuring Apache Web Server

1. Configure the application before installing on your system. This is accomplished with the **configure** command that is provided with the Apache server software distribution. The **configure** command program assesses your current system resources to ensure that the server will run reliably on your system.

2. After running the **configure** command, compile the software. The Apache software distribution provides the **make** command, which will compile the software according to the configuration information.
3. Now the complied software can be installed. To install the software you can run the **make install** command from the Apache software distribution directory. This command copies the *httpd* daemon and associated configuration files to the installation directory. The Apache software installation is now complete.

Now lets consider how the system process will startup and shutdown. This can be accomplished by copying the **apachectl** file to the /etc/init.d directory. With the **ln** command you can create a link to the operating system startup process so that Apache starts up and shuts down normally and doesn't require manual intervention every time a reboot is performed (*su* to root and then perform this command). Once the Apache server is running, you can *telnet* to port 80 to verify the server process is running.

Another important task before making your web server publicly available is to review and set the parameters in the **httpd.conf** file. There are three separate sections to the file:

1. Global Section
   - Location of files and directories. An example: ServerRoot **/home/http**
   - Network parameters. An example: Network Timeouts
2. Server Section
   - General server parameters. An example: **Port 80**
     (TCP - Port the daemon will be listening to)
   - Access control for file directories
   - Directory listings
   - Apache server logging process (audit control)
   - Web browser response parameters
3. Virtual Hosts
   - Enables you to configure multiple web aliases on the same server

Setting these parameters will ensure you have not compromised the system to intruders.

Other Considerations

As you offer web services from your system you will want to ensure that the setup and configuration of your system is of sufficient performance for the tasks. This is considered a function of *Static Pages* that provide information only. The *Application Services* that provide analytical services demands (memory and CPU) resources and the *Information Store* that allocates space for users to download information to their local machines (e.g., FTP)..

The simplest of scenarios is Static Pages. They usually do not require a large allocation of system resources to host. The Application Services and Information Store may require additional resources in the form of storage, CPU, memory or possibly additional systems. It is important to understand the expected usage and performance of the web server so you can scale the resources appropriately. Nothing is worse than visiting a website that has a poor response time. Because it is sometimes hard to define the use of the web server, you should plan for the worst case (budget and time will dictate capacity).

Glossary and Abbreviations

BSD Berkeley software distribution. BSD refers to the particular version of the UNIX operating system that was developed at and distributed from the University of California at Berkeley. BSD UNIX has been popular and many commercial implementations of UNIX systems are based on or include some BSD code.

Case Sensitive This term defines the operating systems sensitivity to alpha uppercase and lower case characters. (Example: If a password or username was created in all lowercase characters then the operating system would deny access if the user entered their password or username in uppercase characters).

c2t3d1s5 C = controller, t = target, d = disk, s = slice (c2 = controller 2, t3 = target SCSI ID 3, d1 = disk 1, s5 = slice number 5).

Command A command is a specific order from a user to the computer's operating system or to an application to perform a function.

CPU Central processing unit. CPU is an older term for processor and microprocessor, the central unit in a computer containing the logic circuitry that performs the instructions of a computer's programs.

Disk Quota The specific amount of disk space that a defined user or service is allow to use. This is done to prevent over utilization of disk resources.

Disk Mirror (RAID 1) Provides 100% redundancy of data. This means no rebuild is necessary in case of a disk failure, just a copy to the replacement disk. Transfer rates per block is equal to that of a single disk. Under certain circumstances, RAID 1 can sustain multiple simultaneous drive failures. This is the simplest RAID storage subsystem design.

Disk Stripping (RAID 0) RAID 0 implements a striped disk array, the data is broken down into blocks and each block is written to a separate disk drive. I/O performance is greatly improved by spreading the I/O load across many channels and drives. Best performance is achieved when data is striped across multiple controllers with only one drive per controller. No parity calculation overhead is involved. Not a "True" RAID because it is NOT fault-tolerant. The failure of just one drive will result in all data in an array being lost.

Disk Stripping with Parity Bit (RAID 3 to 6) Each entire data block is written on a data disk; parity for blocks in the same rank is generated on Writes, recorded in a distributed location and checked on Reads. RAID Level 5 requires a minimum of 3 drives to implement and is the most common RAID with Parity Bit used.

DVD Digital versatile disc. DVD is an optical disc technology that is expected to rapidly replace the CD-ROM disc (as well as the audio compact disc) over the next few years. The digital versatile disc (DVD) holds 4.7 gigabyte of information on one of its two sides, or enough for a 133-minute movie. The DVD can hold more than 28 times as much information as a CD-ROM.

Encryption The conversion of data (i.e., password) into a form called ciphertext, which cannot be easily understood. Decryption is the process of converting encrypted data back into its original form, so it can be understood.

FTP File transfer protocol. FTP is a standard Internet protocol and is the simplest way to exchange files between computers on the Internet.

GUI A graphical (rather than purely textual) user interface to a computer.

HTML Hypertext markup language. HTML is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page.

HTTP Hypertext transfer protocol. HTTP is the set of rules for exchanging files (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

ICMP Internet control message protocol. ICMP is a message control and error-reporting protocol between a host server and a gateway to the Internet.

IP Internet protocol. IP is the method or protocol by which data is sent from one computer to another on the Internet/network. Each computer (known as a host) on the Internet/network has at least one IP address that uniquely identifies it from all other computers on the Internet.

IP Address An IP address has two parts: the identifier of a particular network on the network and an identifier of the particular device (which can be a server or a workstation) within that network.

IP Packet A packet is the unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network. When any file (e-mail message, HTML file, Graphics Interchange Format file, Uniform Resource Locator request, and so forth) is sent from one place to another on the Internet, the Transmission Control Protocol (TCP) layer of TCP/IP divides the file into "chunks" of an efficient size for routing. Each of these packets is separately numbered and includes the Internet address of the destination. The individual packets for a given file may travel different routes through the Internet. When they have all arrived, they are reassembled into the original file (by the TCP layer at the receiving end).

Kerberos Is a secure method for authenticating a request for a service in a computer network.

LINUX Is an UNIX-like operating system that was designed to provide personal computer users a free or very low-cost operating system comparable to traditional and usually more expensive UNIX systems.

Man Pages These are the help files associated with the Solaris™ operating system.

Multi-User System An operating system (e.g. UNIX or LINUX) that will support multiple user access at the same time. (Example: UNIX Operating System)

NFS Network file system. NFS is a client/server application that lets a computer user view and optionally store and update file on a remote computer as though they were on the user's own computer.

OSI Model Is a standard reference model for communication between two end users in a network. It is used in developing products that utilize network technology.

RAID Redundant array of inexpensive disks. This is a way of storing the same data in different places (thus, redundantly) on multiple hard disks. By placing data on multiple disks, I/O operations can overlap in a balanced way, improving performance. Since multiple disks increases the mean time between failure (MTBF), storing data redundantly also increases fault-tolerance.

Parameter Is an item of information such as: a name, a number, or a selected option that is passed to a program by a user or another program.

Reboot The process of restarting a computer operating system.

Power-down The process of shutting down a computer operating system to the point where power can be disconnected from the hardware.

RPC Remote procedure call. RPC is a protocol that one program can use to request a service from a program located in another computer in a network without having to understand network details.

SCSI Small computer system interface. SCSI is a set of ANSI standard electronic interfaces that allow personal computers to communicate with peripheral hardware such as disk drives, tape drives, CD-ROM drives, printers, and scanners faster and more flexibly than previous interfaces.

Server In general, a server is a computer that provides services to other computers and or multiple users.

Signal Messages sent to applications or the operating system to control system processes (i.e., start, stop, suspend, etc).

Single User System A computer operating system that supports only one user at a time (Example: Microsoft® Windows Desktop Operating System).

SNMP Simple network management protocol. SNMP is the protocol governing network management and the monitoring of network devices and their functions.

TCP Transmission control protocol. TCP is a set of rules (protocol) used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet or network.

Telnet Telnet is the way you can access someone else's computer, assuming they have given you permission.

TFTP Trivial file transfer protocol. TFTP is an Internet software utility for transferring files that is simpler to use than the File Transfer Protocol (FTP) but less capable. It is used where user authentication and directory visibility are not required.

UDP User datagram protocol. UDP is a communications protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP).

UFS The default disk based file system for the Sun Operating System 5.x environment.

UNIX is an operating system that originated at Bell Labs in 1969 as an interactive time-sharing system. Ken Thompson and Dennis Ritchie are considered the inventors of UNIX.

URL Uniform resource locator. URL is the address of a file (resource) accessible on the Internet. The type of resource depends on the Internet application protocol.

Suggested Reading

Nemeth, E., Snyder, G., Seebass, S., and Hein, T. R. (2000) UNIX System Administration Handbook, 3rd ed., Prentice Hall, Upper Saddle River, NJ.

Dimambro, B. (2000) Practical Solaris Administration: System Configuration and File Systems, MacMillan Technical Publishing, Indianapolis, IN.

Winsor, S. (1998) Advanced System Administrator's Guide, New Riders Publishing, Indianapolis, IN.

Watters, P. A. and Sumser, J. (2002) Solaris 8 Administrator's Guide, O'Reilly and Associates, Sebastopol, CA.

Howard, J. S. and Deeths, D. (2001) Boot Disk Management: A Guide for the Solaris Operating Environment, Prentice Hall, Upper Saddle River, NJ.

Winsor, J. (2002) Solaris Management Console Tools, Prentice Hall, Upper Saddle River, NJ.

Mulligan, J. P. (2001) Solaris 8 Essential Reference, New Riders Publishing, Indianapolis, IN.

Stringfellow, S., Barton, M., and Klivansky, M. (2000) Backup and Restore Practices for Sun Enterprise Servers, Prentice Hall, Upper Saddle River, NJ.

B. Managing Bioinformatics Tools

16 Installing Bioinformatics Software in a Server-Based Computing Environment

Brian Fristensky

Introduction

To support a diverse institutional program of genomics projects, it is often necessary to have an equally diverse and comprehensive software base. Although programs may come from many sources, it is important to make them easily accessible to the user community on a single computing platform. This chapter will outline the strategies for installing programs for a server-based molecular biology software resource, accessed by a large user base. It is assumed that the reader is familiar with basic UNIX commands and concepts, as described in the previous chapters. The approaches discussed here are implemented in the BIRCH system (*see* Website: http://home.cc.umanitoba.ca/~psgendb), but are generally applicable to any centralized multiuser software installation. The important parts of the process are described in either program documentation or UNIX documentation. The tricks and conventions that help to simplify the installation process will also be highlighted. This should give the novice an idea of what to expect before wading through the documentation.

Considerations

There are five guiding principles for installation and use that should be applied to help ensure a smooth operation.

1. Any user should be able to run any program from any directory simply by typing the name of the program and arguments. It should not be necessary to go to a specific directory to run a program.
2. System administration should be kept as simple as possible. This saves work for the Bioadmin[1], as well as increasing the likelihood that things will function properly.
3. Avoid interruption of service during installation and testing.
4. The Bioadmin should never have to modify individual user accounts.
5. Even if you have root access, do most of your work on a regular user account. Log in as root only when necessary.

[1] Since bioinformatics software may be installed by a specialist other than UNIX system staff, the term *Bioadmin* will refer to the person installing and maintaining bioinformatics software, distinct from system administrators.

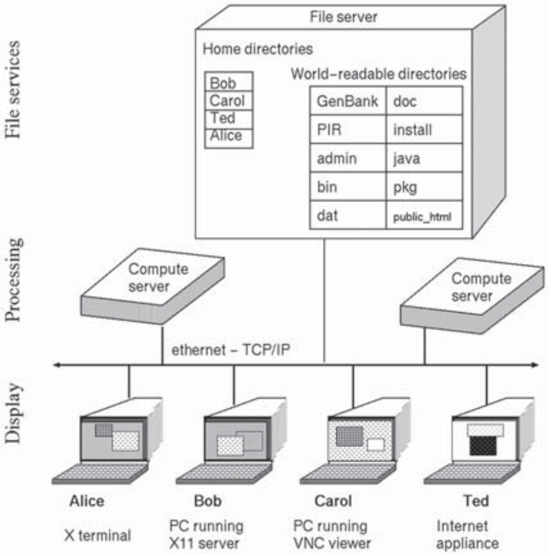


Fig. 1. Network-centric computing.

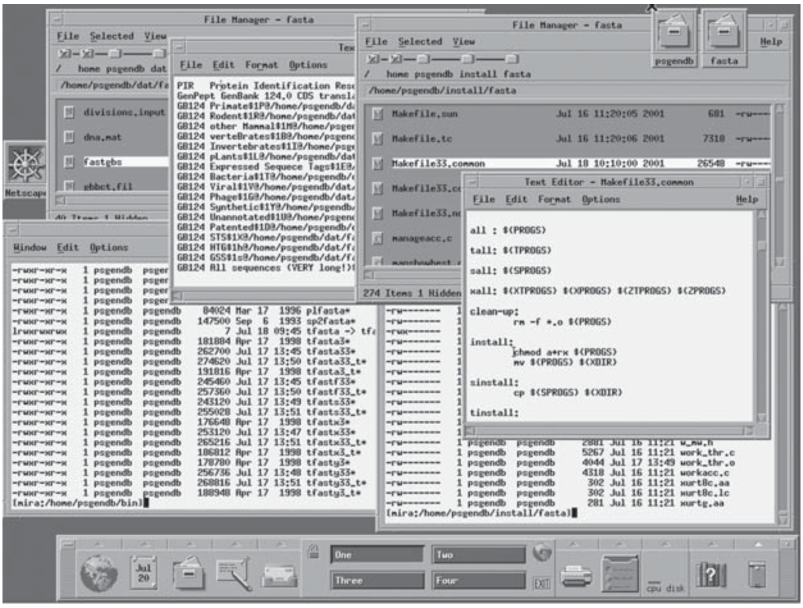
The Networked Computing Environment

The casual computer user learns only a narrow computing model: the PC model. PCs are based on the idea that each person has their own computer that is completely self-contained, with all hardware, software, and data residing physically on the desktop. Provisions for multiple users on a single machine (e.g., separate home directories, user accounts, file permissions) may exist, but are seldom taken into account by PC software developers. Each PC becomes a special case with special problems. The work of administration grows with the number of computers. Software has to be purchased and independently installed for each machine. Security and backup are often not practiced.

UNIX greatly simplifies the problem of computing with a network-centric approach, in which any user can do any task from anywhere. Figure 1 illustrates computing in a network-centric environment. All data and software reside on a file server, which is remotely mounted to one or more identically configured compute servers. Programs are executed on a compute server, but displayed at the user's terminal or PC. Regardless of whether one logs in from an X11 terminal, a PC running an X11 server, a PC using the VNC viewer (*see* Chapters 13 and 17) or an internet appliance, the user's desktop screen looks the same and opens to the user's $HOME directory. Consequently, any user can do any task from any device, anywhere on the Internet.

Leveraging the Multi-Window Desktop

The installation process involves moving back and forth among several directories, which is most effectively accomplished by viewing each directory in a separate window. One of the issues that makes the typical PC desktop awkward to use is the *one window owns the screen* model. In MS-Windows, most applications default to



```
Makefile.sun              Jul 16 11:20:05 2001     681 -rw---
Makefile.tc               Jul 16 11:20:06 2001    7318 -rw---
Makefile33.common         Jul 18 10:10:00 2001   26548 -rw---
```

```
PIR   Protein Identification Reso
GenPept GenBank 124.0 CDS transla
GB124 Primate$1P@/home/psgendb/da
GB124 Rodent$1R@/home/psgendb/dat
GB124 other Mammal$1M@/home/psgen
GB124 verteBrates$1B@/home/psgend
GB124 Invertebrates$1I@/home/psge
GB124 pLants$1L@/home/psgendb/dat
GB124 Expressed Sequece Tags$1E@/
GB124 Bacteria$1T@/home/psgendb/c
GB124 Viral$1V@/home/psgendb/dat/
GB124 Phage$1G@/home/psgendb/dat/
GB124 Synthetic$1Y@/home/psgendb/
GB124 Unannotated$1U@/home/psgend
GB124 Patented$1D@/home/psgendb/c
GB124 STS$1X@/home/psgendb/dat/fa
GB124 HTG$1h@/home/psgendb/dat/fa
GB124 GSS$1s@/home/psgendb/dat/fa
GB124 All sequences (VERY long!)$
```

```
all : $(PROGS)

tall: $(TPROGS)

sall: $(SPROGS)

xall: $(XTPROGS) $(XPROGS) $(ZTPROGS) $(ZPROGS)

clean-up:
        rm -f *.o $(PROGS)

install:
        chmod a+rx $(PROGS)
        mv $(PROGS) $(XDIR)

sinstall:
        cp $(SPROGS) $(XDIR)

tinstall:
```

```
-rwxr-xr-x   1 psgendb  psger
-rwxr-xr-x   1 psgendb  psger
-rwxr-xr-x   1 psgendb  psger
-rwxr-xr-x   1 psgendb  psger
-rwxr-xr-x   1 psgendb  psgendb   84024 Mar 17  1996 plfasta*
-rwxr-xr-x   1 psgendb  psgendb  147500 Sep  6  1993 sp2fasta*
lrwxrwxrwx   1 psgendb  psgendb       7 Jul 18 09:45 tfasta -> tfa
-rwxr-xr-x   1 psgendb  psgendb  181884 Apr 17  1998 tfasta3*
-rwxr-xr-x   1 psgendb  psgendb  262700 Jul 17 13:45 tfasta33*
-rwxr-xr-x   1 psgendb  psgendb  274620 Jul 17 13:50 tfasta33_t*
-rwxr-xr-x   1 psgendb  psgendb  191816 Apr 17  1998 tfasta3_t*
-rwxr-xr-x   1 psgendb  psgendb  245460 Jul 17 13:45 tfastf33*
-rwxr-xr-x   1 psgendb  psgendb  257360 Jul 17 13:50 tfastf33_t*
-rwxr-xr-x   1 psgendb  psgendb  243120 Jul 17 13:49 tfasts33*
-rwxr-xr-x   1 psgendb  psgendb  255028 Jul 17 13:51 tfasts33_t*
-rwxr-xr-x   1 psgendb  psgendb  176648 Apr 17  1998 tfastx3*
-rwxr-xr-x   1 psgendb  psgendb  253120 Jul 17 13:47 tfastx33*
-rwxr-xr-x   1 psgendb  psgendb  265216 Jul 17 13:51 tfastx33_t*
-rwxr-xr-x   1 psgendb  psgendb  186812 Apr 17  1998 tfastx3_t*
-rwxr-xr-x   1 psgendb  psgendb  178780 Apr 17  1998 tfasty3*
-rwxr-xr-x   1 psgendb  psgendb  256736 Jul 17 13:48 tfasty33*
-rwxr-xr-x   1 psgendb  psgendb  268816 Jul 17 13:51 tfasty33_t*
-rwxr-xr-x   1 psgendb  psgendb  188948 Apr 17  1998 tfasty3_t*
[mira:/home/psgendb/bin]
```

```
-rw-------   1
-rw-------   1
-rwx------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1
-rw-------   1 psgendb  psgendb    2881 Jul 16 11:21 w_mw.h
-rw-------   1 psgendb  psgendb    5267 Jul 16 11:21 work_thr.c
-rw-------   1 psgendb  psgendb    4044 Jul 17 13:49 work_thr.o
-rw-------   1 psgendb  psgendb    4318 Jul 16 11:21 workacc.c
-rw-------   1 psgendb  psgendb     302 Jul 16 11:21 xurt8c.aa
-rw-------   1 psgendb  psgendb     302 Jul 16 11:21 xurt8c.lc
-rw-------   1 psgendb  psgendb     281 Jul 16 11:21 xurtg.aa
[mira:/home/psgendb/install/fasta]
```

Fig. 2. Leveraging the multiwindow desktop. In the example, the environment variable $DB, which identifies the root directory for bioinformatics software, is set to /home/psgendb. Clockwise from top left: Double clicking on **fastgbs** in $DB/doc/fasta opens up a list of locations of database files in a text editor. Similarly, Makefile33.common has been opened up from $DB/install/fasta. This file contains Makefile commands that are compatible with all operating systems. A terminal window at lower right is used for running commands in $DB/install/fasta, while another terminal window at bottom left is used for running commands in $DB/bin. At bottom, the CDE control panel shows that the current screen, out of four screens available, is screen One.

Note: To get the C-shell to display the hostname and current working directory in the prompt, include the following lines in cshrc.source or your .cshrc file:

```
set HOSTNAME = ‘hostname‘
set prompt=”{$HOSTNAME”:”$cwd”}
alias cd ‘cd \!*; set prompt=”{$HOSTNAME”:”$cwd”}’
alias popd ‘popd \!*; set prompt=”{$HOSTNAME”:”$cwd”}’
alias pushd ‘pushd \!*; set prompt=”{$HOSTNAME”:”$cwd”}’
```

take up the entire screen. One moves between applications using the task bar. Although it is possible to resize windows so that many can fit on one screen, this is seldom done in MS-Windows. On Macintosh, even when multiple windows are present, they depend on the menu at the top of the screen. This requires the user to first select a window by clicking on it, and then to choose an item from the menu at the top of the screen. Even worse, the menus often look almost identical from program to program, so that it is not obvious when the focus has shifted to a new application.

On UNIX desktops, menus are found within the windows themselves. This decreases the amount of distance the eye has to cover. Focus moves with the mouse and does not need to be switched with the taskbar. The user simply moves from one window to another to work. Because UNIX tends to be oriented towards multiple windows, UNIX users tend to favor larger monitors. More screen real estate means more space for windows. The screen in Fig. 2 appears crowded because it was generated at

Table 1
Sources of Free Downloadable Software

Source	*URL*
IUBio Archive	http://iubio.bio.indiana.edu/
EMBOSS Software Suite	http://www.uk.embnet.org/Software/EMBOSS/
Open Source Bioinformatics Software	http://bioinformatics.org/
Linux for Biotechnology	http://www.randomfactory.com/lfb/lfb.html
Sanger Center Software	http://www.sanger.ac.uk/Software/
Staden Package	http://www.mrc-lmb.cam.ac.uk/pubseq/
NCBI FTP site	http://www.ncbi.nlm.nih.gov/Ftp/index.html
PHYLIP Phylogeny software	http://evolution.genetics.washington.edu/phylip.html
BIRCH, FSAP, XYLEM,GDE	http://home.cc.umanitoba.ca/~psgendb/downloads.html
FASTA package	ftp://ftp.uva.edu/pub/fasta/
Virtual Network Computing (VNC)	http://www.uk.research.att.com/vnc/

1024 × 768 resolution. Although this is a common resolution for PCs, the UNIX community tends to work with larger monitors, at 19" diagonal or larger, running at 1280 × 1000 or higher resolution. To provide further real estate, UNIX desktops such as CDE, KDE, and GNOME support switching between several desktops at the push of a button. Use of multiple windows during an installation is illustrated in Figure 2.

Finding and Downloading Software

Table 1 has a short and by no means exhaustive list of sites where freely available sequence analysis software can be downloaded. USENET newsgroups such as bionet.software contain announcements of new software and updates, as well as discussions on molecular biology software.

Usually, software is downloaded as a directory tree packed into a single archive file in various formats. Generally, files in these formats can recreate the original directory tree containing source code, documentation, data files, and often, executable binaries. Usually, the first step is to uncompress the file and then recreate the original directory. For example, the fasta package comes as a shell archive created using the **shar** command. Because you do not know in advance whether the archive contains a large number of individual files or a single directory containing files, it is always safest to make a new directory in which to recreate the archive, using the following commands:

mkdir fasta	create new directory
mv fasta.shar fasta	move fasta.shar into the new directory
cd fasta	go into the fasta directory
unshar fasta.shar	extract files from fasta.shar

Table 2 lists some of the most common archive tools and their usage.

Two goals when installing software are to 1) avoid interruption of service for users during installation and testing and 2) having the option of deleting programs after

Table 2
Archive Commands

File extension	*Utility*	*Archive command*	*Unarchive command*
.tar	UNIX tar	To create a tar file from a directory called source: **tar cvf source.tar source**	To recreate the directory: **tar xvf source.tar**
.zip	ZIP	To create a compressed archive file called source.zip: **zip source source**	To recreate the directory: **unzip source**
.shar	UNIX shar	To create a shar file from a directory named source: **shar source > source.shar**	To recreate the directory: **unshar source.shar** or **chmod u+x source.shar** or **sh source.shar**[a]
.gz	GNU zip	To compress a file with: **gzip source > source.gz**	To recreate the directory: **gunzip source**
.Z	compress	To compress source.tar: **compress source.tar**	To uncompress source.tar.Z: **uncompress source.tar.Z**
.uue	uuencode	To encode source.tar.Z using ASCII characters: **uuencode source.tar.Z source.uue**	To recreate the original binary file: **uudecode source.uue**

[a] .shar files are actually shell scripts that can be executed to recreate the original directory.

evaluation. For example, a separate directory called *install* could hold separate directories for each package during the installation.

Understand the Problem Before You Begin

For many standard office tasks, it is possible to get by without ever reading the documentation. In molecular biology, the task itself often has enough complexity that it may not be possible to simply launch, point, and click. In practice, it is almost always faster to read the documentation before trying to install. Each program will have installation instructions. These will let you know about important options like where the final program files will reside and which environment variables must be set.

Reading the documentation at this stage gives you a chance to learn more about what the program does and to decide if it is really what you need. This weeding out phase can save a lot of unnecessary compiling, organizing, and testing.

Compilation

Programs distributed as source code, for which no binaries are available, will require compilation and linking steps. Although these procedures vary somewhat with the language in which they are written, most of the common packages use protocols of the C and C++ family of languages. In addition to source files (.c), code items such as

Table 3
Common File Types and File Extensions

File extension	*File type*
.c	C source code
.h	C header
.o	Compiled object file
no extension	Executable binary file
.1, .l	UNIX manual page
.makefile, .mak	Makefile

type definitions, which need to be shared, are found in header files (.h). When you compile, the code from the header files is inserted into C code, and the .c file translated to machine code, which is written as object modules (.o). Next, the compiler calls a linker, which links object modules into a final executable file. In most programs, object modules from standard libraries (e.g., Tcl/Tk) are also linked. These are typically linked dynamically, meaning that only a reference to the libraries is made and the actual library modules are loaded each time the program is run. Consequently, dynamic linking saves disk space. However, when a program depends on libraries that may not be found on all systems, static linking can be accomplished, in which object modules are written to the final executable code file. Static linking favors portability at the expense of disk space. A short list of the types of files frequently encountered during installation appears in Table 3.

Virtually all scientific program packages automate these procedures using the *make* program. The *make* program reads a Makefile, containing compilation, linking, and installation options. For cross-platform compatibility, it is common to include separate Makefiles for each platform (e.g., SGI, Linux, Windows, Solaris). For example, the fasta package has a file called Makefile.sun for Solaris systems. Copy Makefile.sun to Makefile, and edit Makefile as needed for your system. At the beginning of the Makefile, variables are often set to specify the final destinations for files. On our system, fasta's Makefile would be edited to change the line reading **XDIR = /seqprog/sbin/bin to XDIR = /home/psgendb/bin.** Because this directory is in the $PATH for all BIRCH users, the new programs become available to all users as soon as the files are copied to this location.

Typing **make** executes the commands in Makefile, compiling and linking the programs. It is best to run make in a terminal window that supports scrolling, so that all warning and error messages can be examined. This is particularly important because one can then copy error messages to a file to provide the author of the program with a precise description of the problem. If the authors do not receive this feedback, the problems do not get fixed. However feedback must be precise and detailed.

If **make** is successful, executable binary files, usually with no extension, are written to the target directory. This may or may not be the current working directory. Many Makefiles require you to explicitly ask for files to be copied to the destination directory by typing **make install**.

In some cases, testing can be carried out at this point, particularly if a test script is included with the package. In the fasta package several test scripts are found. For example, **./test.sh** will run most of the fasta programs with test datafiles.

Note: "./" forces the shell to look for test.sh in the current working directory. Unless "." was in your $PATH, it would not be found. However, it is generally considered insecure to include "." in your $PATH.

It is important to check test scripts to see where they look for executable files. For example, if the script sets $PATH to "." (the current directory), or if programs are executed with a statement such as **./fasta33**, then the shell will look for an executable file in the current directory. If the directory for the executable file is not explicitly set, the shell will search all directories in your $PATH to find an executable file. This could either result in a *Command not found* message, or if earlier copies of the programs were already installed, older programs will execute, not the newly compiled programs.

Installation

In the BIRCH system, all files and directories are found in a world-readable directory specified by the $DB environment variable. Thus, $DB/bin, $DB/doc, and $DB/dat refer to directories containing executable binaries, documentation, and datafiles used by programs, respectively, as summarized below.

$DB/bin

Although $DB/bin could in principle be set to refer to /usr/local/bin, it is probably best to keep the entire $DB structure separate from the rest of the system. This approach has the advantage that the Bioadmin need not have root privileges. All files in $DB/bin should be world-executable.

One practice for managing program upgrades is to create a symbolic link to point to the current *production* version of the program. For example, a link with the name *fasta* might point to fasta3:

```
lrwxrwxrwx 1 psgendb psgendb 6 Jul 18 09:45 /home/psgendb/bin/fasta3 -> fasta*
```

To upgrade to fasta33:

```
rm fasta
ln -s fasta33 fasta
lrwxrwxrwx 1 psgendb psgendb 6 Jul 18 09:45 /home/psgendb/bin/fasta33 -> fasta*
```

Aside from giving users a consistent name for the current most recent version of the program, this type of stable link eliminates the need to modify other programs that call the upgraded program.

$DB/doc

Documentation files for software should be moved to this directory. Ideally, the complete contents of this directory should be Web-accessible. Where a program or package has more than one documentation file, create a separate subdirectory for each program. All files should be world-readable.

$DB/dat

A program should never require that ancillary files such as fonts or scoring matrices be in a user's directory. These should always be centrally installed and administered, transparently to the user. Datafiles required for programs, such as scoring matrices, lists of restriction enzymes, and so forth should be moved to this directory. Generally, each program should have its own subdirectory. All files should be world-readable.

$DB/admin

This directory contains scripts and other files related to software administration. First time BIRCH users run the *newuser* script, to append a line to the user's .login file:

source /home/psgendb/admin/login.source

and to the .cshrc file:

source /home/psgendb/admin/cshrc.source

These files respectively contain commands that are executed when the user first logs in, and each time a program is started. All environment variables, aliases, and other settings required to run these programs are set in these files. Having run *newuser* once, a user should never have to do any setup tasks to be able to run new or updated programs. When a new program is added, the environment variables and aliases needed are specified in cshrc.source, and therefore become immediately available to the user community. The net effect is that the Bioadmin should never have to go to each user's account when a new program is installed.

Where programs require first-time setup, such as a configuration file being written to the user's $HOME directory, the program should be run from a wrapper script that checks for the presence of that file. If the file is not present, the script writes a default copy of the file to $HOME. The user should never have to explicitly run a setup script before using a program.

All directories that are to be accessible to users must be world searchable (world-executable), as well as world-readable. For example, to allow users to read files in $DB/doc/fasta, both $DB/doc and $DB/doc/fasta must be world executable. Use:

chmod a+rx $DB/doc/fasta

Special Considerations for Complex Packages

Some packages come as integrated units whose components can not be moved out of their directory structure to $DB/bin, $DB/dat, and $DB/doc. Packages of this type are installed in $DB/pkg. A good case in point is the Staden Package (*see* Table 1). The BIRCH login.source file contains the following lines:

Initialize Staden Package Settings
setenv STADENROOT $DB/pkg/staden
source $STADENROOT/staden.login

that cause the commands in staden.login to be executed when the user logs in. This script in turn sets several environment variables referencing files in the $STADENROOT directory. As well, login.source adds $DB/pkg/staden/solaris-bin to the $PATH. This illustrates that there are sometimes no simple solutions. On one hand, it would be desirable to simply copy all Staden binaries into $DB/bin. However this has the effect of making it difficult to identify the origin of specific programs in

$DB/bin as being part of the Staden Package. On the other hand, symbolic links from $DB/bin to each of the programs in $DB/pkg/staden/solaris-bin would provide a compromise. This would require that links be individually maintained.

Documentation and data can be linked more easily in $DB/doc. The following command:

```
ln -s $STADENROOT/doc staden
```

creates a link to the Staden documentation directory, while in $DB/dat, and

```
ln -s $STADENROOT staden
```

creates a link to the main Staden directory, from which the user can find several directories with sample datafiles. In this fashion, the $DOC and $DAT directories appear to contain *staden* subdirectories, whose contents are physically located elsewhere.

On Linux systems, complex packages are maintained using programs such as Red Hat Package Manager (RPM). RPM automates package installation, often requiring no user input. As files are copied to their destinations, their locations are recorded in the RPM database (/usr/lib/rpm). In addition to installation, RPM automates package updating, verification and de-installation. Tools such as RPM make it possible to install software in system directories such as /usr/local/bin or /usr/bin without making these directories unmanageable. The one disadvantage is that installation in system directories can only be accomplished with root permissions.

Special Considerations for Java Applications

Java applications should be installed in a central location, such as $DB/java. Ideally, all that should be required is the inclusion of $DB/java in the $CLASSPATH environment variable. The Java Virtual Machine (JVM) would search this location at runtime. However, the precise commands needed to launch an application vary, so that no single solution exists. For example, some applications are completely contained in a single .jar file, while others require a complex directory structure containing large numbers of objects and datafiles. Consequently, Java applications should be launched from wrappers: short scripts that reside in $DB/bin and call the application. For example, a script called $DB/bin/readseq runs the Java implementation of readseq (available from IUBio, *see* Table 1):

```
#!/bin/csh
# UNIX script file to run command line readseq

# Full path must be specified to enable us to
# launch readseq from any directory.
setenv CLASSPATH $DB/java/readseq/readseq.jar

# $argv passes command line arguments from the wrapper
# to the application
java -cp $CLASSPATH run $argv
```

Thus, typing **readseq** launches the wrapper, which in turn launches the Java readseq application. Readseq also has a method called *app* which runs readseq in a graphic interface. To run in this mode, the Xreadseq wrapper has the following line:

```
java -cp $CLASSPATH app
```

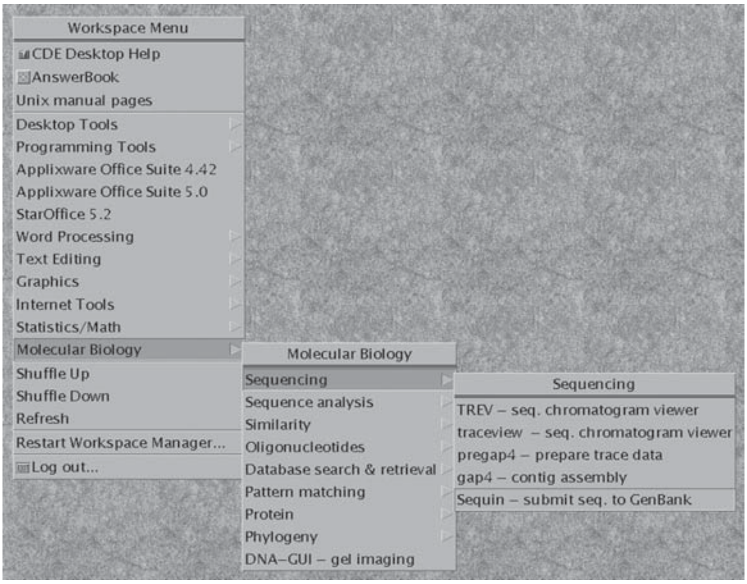


Fig. 3. The CDE Workspace Menu.

Calling Programs from a Graphic Front-End

Several options exist for unifying a large software base with a graphic front end. Programs such as GDE (*see* Table 1), SeqLab from the GCG package (*see* Website: http://www.accelrys.com), or SeqPup (*see* Table 1, IUBio) allow the Bioadmin to add external programs to their menus, as specified in easy-to-edit configuration files. In general, the user selects one or more sequences to work with, and then chooses a program from the main menu. A window pops up, allowing parameters to be set. The front end then generates a UNIX command to run the program with these parameters, using the selected sequences.

In many cases, it is best for the front end to call a wrapper that verifies and checks the parameters and sequences, then executes the program. This is especially important for programs from packages such as FSAP or PHYLIP (*see* Table 1), which operate through text-based interactive menus. If a prompt does not receive a valid response, programs of this type may go into an infinite loop that prompts for a response.

Launching Programs from the Workspace Menu

The *workspace* menu is yet another avenue through which users can find programs. Figure 3 shows a CDE workspace menu organized categorically. At the highest level are the main categories of programs, including standalone items for office packages. The *molecular biology* menu is further divided into submenus. For example, the *Sequencing* submenu contains programs that together cover all steps in the sequencing process, including reading the raw chromatograms, vector removal, contig assembly, and submission to GenBank. The downside of the workspace menu is that it is incomplete, as command line applications, cannot be launched from the workspace menu.

One should be able to launch GUI applications from the workspace menu, defaulting to the user's $HOME directory. In the CDE desktop, this is specified in the .dt directory. Most recent UNIX desktops, such as CDE 1.4 and GNOME use tree-structured directories to define the structure of the workspace menu. Again, it is important to avoid having to update individual user accounts. The easiest way is to create a directory for molecular biology programs on the Bioadmin's account, and have new users run a script creating a symbolic link from their workspace menu directory (e.g., .dt/Desktop) to the Bioadmin's directory. Subsequently, all updates to the Bioadmin's menu will become available to all users.

Testing

Testing should not be carried out using the account that owns the programs (e.g., root). Testing should always be done in a regular user account. One reason is that testing on a user account will uncover incorrect permissions. This is likely the single most common installation error. At the same time, it is probably also best to test in a subdirectory, rather than in the $HOME directory, to fully demonstrate that the program can be run from anywhere.

Using VNC (see Table 1), one can easily eliminate login/logout cycles between your Bioadmin account and your user account. The vncserver is an X11 server that runs on a networked UNIX host. It creates an X11 screen in memory. To display that screen, run vncviewer on your desktop machine. The complete UNIX desktop appears in a window. For example, the entire screen shown in Fig. 2 was run in a vncviewer window. Thus, switching back and forth between the user desktop and the Bioadmin desktop is as easy as switching between windows, facilitating rapid test-modify cycles. (See Chapter 17 for additional VNC insights.)

Installation Checklist

Before announcing updates or new programs, go through the package in a user account, checking the following:

- All files world-readable (**chmod a+r filename**)
- All binaries world-executable (**chmod a+x filename**)
- All directories world-searchable (**chmod a+x directoryname**)
- Environment variable set in cshrc.source
- Documentation and datafiles updated

Although installation should result in a *finished product*, there are often bugs that need to be worked out as the package or program gets used. At this point, the user base is probably the best group of testers, becuase they will make mistakes, and they will try a wider range of data than the Bioadmin would try.

Acknowledgments

Thanks to the Academic Computing and Networking staff at the University of Manitoba for UNIX system support. This work was made possible in part through hardware provided by the Sun Academic Equipment Grants Program.

Glossary and Abbreviations

Front end A user interface that executes other programs.

GUI Graphic user interface.

JVM Java virtual machine. JVM emulates a computer architecture in software. Java programs are compiled into pseudocode (P-code) which is executed by JVM. Consequently, compiled Java code runs on any computer for which JVM is available.

Thin client A display device that draws a screen as specified by the X-server. Thin clients are almost always diskless. Execution of the program occurs on the host, and the results are displayed on the client. X-terminals, SunRay(r) Internet Appliances, and even PC's running X11 server are all examples of thin clients.

Wrapper A script that automates startup tasks required for execution of a program. For example, wrappers often check for critical files and set environment variables.

X11 (X windows) Protocols and libraries for Unix graphic interfaces. The X-server (X) is a program that executes X11 calls from X-applications, and draws the specified screen.

Suggested Readings

Introduction

Fristensky, B. (1999) Building a multiuser sequence analysis facility using freeware, in: Bioinformatics Methods and Protocols, (Misener, S. and Krawetz, S., eds.), Humana Press, Totowa, NJ, pp. 131–145.

Sobell, M. G. (1995) A Practical Guide to the UNIX System, Benjamin/Cummings Publishing Co., Redwood City, CA.

The Networked Computing Environment

Fristensky, B. (1999) Building a multiuser sequence analysis facility using freeware, in: Bioinformatics Methods and Protocols, (Misener, S. and Krawetz, S., eds.), Humana Press, Totowa, NJ, pp. 131–145.

Compilation

Pearson, W. R. (1999) Flexible sequence similarity searching with the FASTA3 program package, in: Bioinformatics Methods and Protocols, (Misener, S. and Krawetz, S., eds.) Humana Press, Totowa, NJ, pp. 185–219.

Installation

Felsenstein J. (1989) PHYLIP Phylogeny Inference Package, Cladistics 5, 164–166.

Fristensky, B., Lis, J. T., and Wu, R. (1982) Portable microcomputer software for nucleotide sequence analysis, Nucl. Acids Res. 10, 6451–6463.

Fristensky, B. (1986) Improving the efficiency of dot-matrix similarity searches through use of an oligomer table, Nucl. Acids Res. 14, 597–610.

Fristensky, B. (1999) Building a multiuser sequence analysis facility using freeware, in: Bioinformatics Methods and Protocols, (Misener, S. and Krawetz, S., eds.), Humana Press, Totowa, NJ, pp. 131–145.

Smith, S., Overbeek, R., Woese, C. R., Gilbert, W., and Gillevet, P. M. (1994) The Genetic Data Environment: an expandable GUI for multiple sequence analysis, Comp. Appl. Biosci. 10, 671–675.
(*see* Website: http://megasun.bch.umontreal.ca/pub/gde/)

Staden R., Beal, K. F., and Bonfield, J. K. (1999) The Staden Package, in: Bioinformatics Methods and Protocols , (Misener, S. and Krawetz, S, eds.), Humana Press, Totowa, NJ, pp. 115–130.

17 Management of a Server-Based Bioinformatics Resource

Brian Fristensky

Introduction

The strategies for managing a server-based molecular biology software resource, accessed by a diverse user community has been discussed in the previous chapters. It assumes that the reader is familiar with basic UNIX commands and concepts. The approaches discussed here are implemented in the BIRCH system (*see* Website: http://home.cc.umanitoba.ca/~psgendb) but are generally applicable to any centralized multiuser software installation.

Most major UNIX distributions now come with graphic tools that simplify many administration tasks. It is therefore realistic to act as your own *sysadmin*. In fact many of the principles discussed are valid in the larger context of a general purpose multiuser system. Although general system administration is a broad field, particular attention should be paid to: daily and weekly backups, both onsite and offsite; security, including rapid installation of security patches; management of user accounts; and disk space to minimize the work and know-how needed on the part of the user. These topics are beyond the scope of this chapter, and are covered extensively in books on system administration, on USENET newsgroups in the comp.* section, and at various HOWTO websites.

The key factors and considerations when implementing the system are:

1. A user base with a diverse set of needs and usually minimal informatics training.
2. A diverse software base, comprised of programs from many authors, in many languages, and in many styles.
3. Documentation written in many formats and styles.
4. A complex networked server system.
5. Limitations of disk space and computing resources.

This chapter builds on the organizational scheme described in the previous chapter. To summarize, the resource is located in a world-readable directory tree referenced by the $DB environment variable. Program binaries, documentation, and ancillary datafiles are located in $DB/bin, $DB/doc, and $DB/dat, respectively. To use the resource, user accounts are set up by running the *newuser* script. This appends a set of configuration commands from $DB/admin/login.source and $DB/admin/cshrc.source to their .login and .cshrc files. The commands in these files are executed when a user logs in or starts a new shell. Thus, as the central configuration is updated, all users have immediate access to the updates. The means to implement this structure are described in the previous chapter.

Managing Documentation

Documentation is the user's entry point into the system. Keeping documentation organized, accessible, and updated accomplishes several tasks. First, it helps to bring out difficulties that users may face when running programs. Second, it forces the Bioadmin to see the software base from the user's perspective. Third, well-organized documentation works to the Bioadmin's advantage, making it easy to refer users to the appropriate documentation, rather than having to answer the same question over and over. While installation of documentation should be straightforward, there are a few considerations for providing a consistent web-accessible documentation library.

HTML

HTML is rapidly becoming the most common format for documentation because of its dynamic capabilities and universal availability. However, it is probably best to keep a local copy of the program documentation on your website, rather than simply linking to the author's website. An author's website will probably describe the most recent version of the software, which may not be installed on your system. If the author stops supporting a software package, he or she may no longer keep documentation on a website. Thus, installation of a local copy of the documentation that was obtained at the time the package was installed is guaranteed to accurately describe the version of the software currently installed.

UNIX Manual Pages

BIRCH has a directory for manual pages called $DB/manl. All files in this directory should be in the form name.l (where l stands for local). In login.source, the line

```
setenv MANPATH $MANPATH\:$DB
```

tells UNIX to look for the manual pages in this directory, as well as in any other directory specified in the system's $MANPATH. For example, to read the documentation for align, the user types **man align**, and the file $DB/manl/align.l will be displayed. To display on the web, UNIX manual pages can be converted to ASCII text by redirecting the output from the **man** command to an ASCII file, e.g.,

```
man fasta > fasta.txt
```

Postscript and PDF

Although PostScript viewers are usually available on most UNIX workstations, **acroread**, the Adobe Acrobat Reader, has been universally adopted. Therefore, it is probably safest to convert postscript files to PDF for web accessibility using **ps2pdf**, e.g., **ps2pdf primer3.ps**, will create a file called primer3.pdf. ps2pdf is included with most UNIX distributions.

ASCII Text

All web browsers can display ASCII text. It should be noted that file extensions such as .txt or .asc are probably best to use, because these are not commonly used by application software. ASCII files with .doc extensions should be changed to some other extension to avoid confusion with Microsoft WORD files.

Word Processor Documents

Import filters are often less than satisfactory. Therefore, when documentation is in a format specific to a word processor such as WordPerfect, StarOffice Writer, Applix Words, or Microsoft Word, it is best to convert it to the PDF format. Some programs can directly save or print to PDF, while others can only print to PostScript. For the latter, convert to PDF using **ps2pdf** as noted earlier.

Communicating with the User Base

Login Messages

Brief announcements can be printed at the user's terminal by including a statement such as **cat ~psgendb/admin/Login_Message** the in login.source, where Login_Message contains a few lines of text with the current announcements. This message is printed in each terminal window.

Web Site Organization

The BIRCH website provides a number of views to the system (*see* Fig. 1). The *New User* section provides documents that describe BIRCH, how to set up accounts, and how to learn the system. The *Documentation* section provides tutorials and other resources for users to develop their informatics skills while getting useful work done. Finally, the complete online documentation is available in the *Software* and *Database* sections, describing the full functionality of the system.

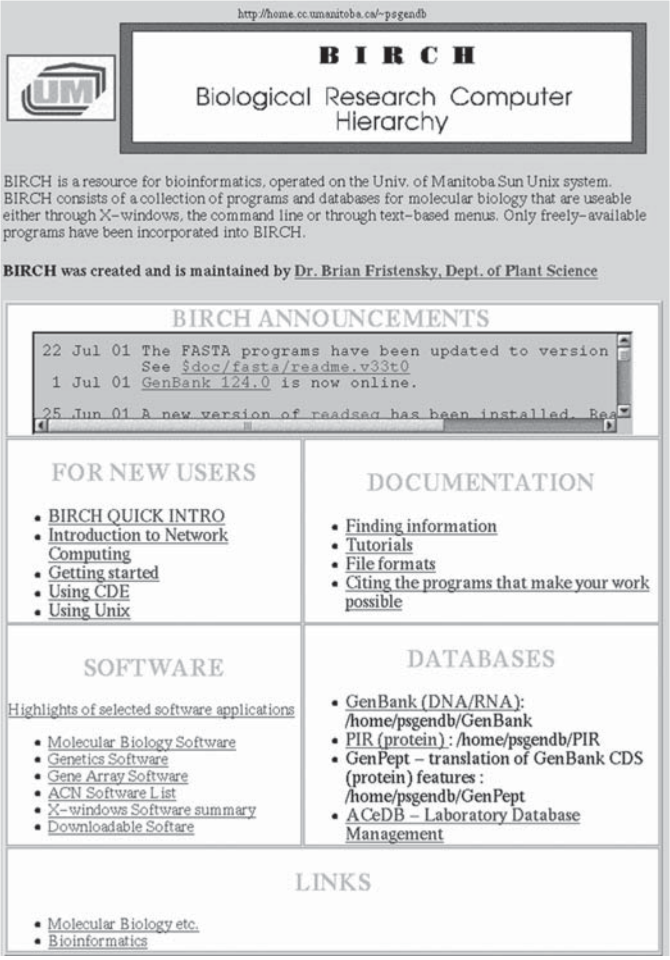
All login messages are archived in the file WHATSNEW.html, which can be viewed in a scrolling window entitled *BIRCH ANNOUNCEMENTS*. This file provides links to more detailed information than appears in login messages, so that even users who have been away from the system for a while won't miss important changes.

Discussion Groups

Although online discussions can be conducted through a mailing list, these often become an annoyance as the number of users increase and the number of lists one subscribes to increases. Most web browsers such as Netscape and Internet Explorer/Outlook Express, as well as third-party applications, can be used to read and participate in discussions on USENET newsgroups. Many users are familiar with worldwide groups, including the bionet.* groups (e.g., bionet.software, bionet.molbio.genearrays). However, it is also possible to have local newsgroups on any system that operates a newsserver, as do most campus UNIX systems. The local news Bioadmin can easily create a group such as *local.bioinformatics* or *local.genomics* that will be accessible to the local user community.

Remote Consultation Using VNC

Remote consultation on UNIX platforms is now greatly enhanced by Virtual Network Computing (VNC). VNC is a package of programs freely distributed by AT&T (*see* Website: http://www.uk.research.att.com/vnc/). In essence, vncserver creates an X11 desktop session on a remote login host, which keeps an image of the screen in



http://home.cc.umanitoba.ca/~psgendb

BIRCH

Biological Research Computer Hierarchy

BIRCH is a resource for bioinformatics, operated on the Univ. of Manitoba Sun Unix system. BIRCH consists of a collection of programs and databases for molecular biology that are useable either through X-windows, the command line or through text-based menus. Only freely-available programs have been incorporated into BIRCH.

BIRCH was created and is maintained by **Dr. Brian Fristensky, Dept. of Plant Science**

BIRCH ANNOUNCEMENTS

```
22 Jul 01 The FASTA programs have been updated to version
          See $doc/fasta/readme.v33t0
 1 Jul 01 GenBank 124.0 is now online.

25 Jun 01 A new version of readseq has been installed. Rea
```

FOR NEW USERS

- BIRCH QUICK INTRO
- Introduction to Network Computing
- Getting started
- Using CDE
- Using Unix

DOCUMENTATION

- Finding information
- Tutorials
- File formats
- Citing the programs that make your work possible

SOFTWARE

Highlights of selected software applications

- Molecular Biology Software
- Genetics Software
- Gene Array Software
- ACN Software List
- X-windows Software summary
- Downloadable Softare

DATABASES

- GenBank (DNA/RNA): /home/psgendb/GenBank
- PIR (protein) : /home/psgendb/PIR
- GenPept – translation of GenBank CDS (protein) features : /home/psgendb/GenPept
- ACeDB – Laboratory Database Management

LINKS

- Molecular Biology etc.
- Bioinformatics

Fig. 1. Organization of web-based documentation on the BIRCH home page (*see* Website: http://home.cc.umanitoba.ca/~psgendb).

memory. A copy of vncviewer, running on a PC or workstation anywhere in the world with a high-speed Internet connection, can display and control the screen as if it were running locally. Figure 2 shows a screen in which a *vncviewer* window is displayed. VNC is available for MS-Windows, Macintosh, and UNIX. The vncviewer can also run as a Java applet in a web browser, so that vncviewer does not have to be installed on the local machine. Thus, regardless of where you are, your UNIX desktop looks and acts the same.

For remote consultation, assume that a user has phoned the Bioadmin with a problem. If it cannot be easily described over the phone, the user changes their VNC password using **vncpw**, and tells the Bioadmin the new password. Next the user starts up a copy of vncserver:

```
vncserver -alwaysshared
New 'X' desktop is mira:8
```

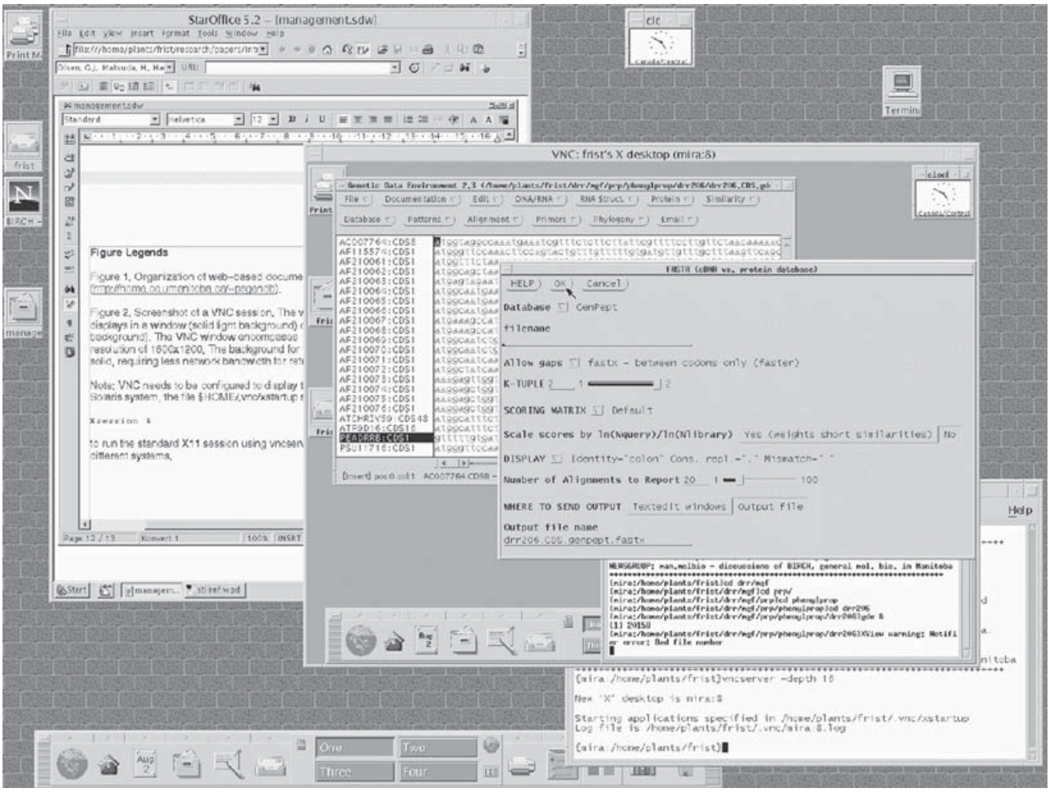

StarOffice 5.2 – [management.sdw]
File Edit View Insert Format Tools Window Help
VNC: frist's X desktop (mira:8)
Genetic Data Environment 2.3 (/home/plants/frist/drr/mgf/prp/phenylprop/drr206/drr206.CDS.gb)
File Documentation Edit DNA/RNA RNA Struct Protein Similarity
Database Patterns Alignment Primers Phylogeny Email
FASTX (cDNA vs. protein database)
HELP OK Cancel
One Two Three Four
Print Manager
frist
BIRCH
management
Terminal
clock
Help

Figure Legends

Figure 1, Organization of web-based docume
(http://home.cc.umanitoba.ca/~psgendb).

Figure 2, Screenshot of a VNC session, The v
displays in a window (solid light background) c
background). The VNC window encompasses
resolution of 1600x1200, The background for
solid, requiring less network bandwidth for ref

Note: VNC needs to be configured to display t
Solaris system, the file $HOME/.vnc/xstartup s

```
Xsession &
```

to run the standard X11 session using vncserv
different systems,

Page 12 / 13 Konvert 1 100% INSRT

Database GenPept

Filename

Allow gaps fastx - between codons only (faster)

K-TUPLE 2 1 2

SCORING MATRIX Default

Scale scores by ln(Nquery)/ln(Nlibrary) Yes (weights short similarities) No

DISPLAY Identity="colon" Cons. repl.="." Mismatch=" "

Number of Alignments to Report 20 1 100

WHERE TO SEND OUTPUT Textedit windows Output file

Output file name
drr206.CDS.genpept.fastx

```
NEWSGROUP: man.molbio - discussions of BIRCH, general mol. bio. in Manitoba
*************************************************************************
[mira:/home/plants/frist]cd drr/mgf
[mira:/home/plants/frist/drr/mgf]cd prp/
[mira:/home/plants/frist/drr/mgf/prp]cd phenylprop
[mira:/home/plants/frist/drr/mgf/prp/phenylprop]cd drr206
[mira:/home/plants/frist/drr/mgf/prp/phenylprop/drr206]gde &
[1] 20150
[mira:/home/plants/frist/drr/mgf/prp/phenylprop/drr206]XView warning: Notifi
er error: Bad file number
```

```
{mira:/home/plants/frist}vncserver -depth 16

New 'X' desktop is mira:8

Starting applications specified in /home/plants/frist/.vnc/xstartup
Log file is /home/plants/frist/.vnc/mira:8.log

{mira:/home/plants/frist}
```

Fig. 2. Screenshot of a VNC session. The vncviewer VNC: frist's X desktop (mira:8) displays in a window (solid light background) on the local desktop (dark brick background). The VNC window encompasses 1024 × 768 pixels, against a screen at a resolution of 1600 × 1200. The background for the mira:8 desktop has been changed to solid, requiring less network bandwidth for refreshing the screen across a network.

Note: VNC needs to be configured to display the user's regular X11 desktop. On our Solaris system, the file $HOME/.vnc/xstartup should contain the line "Xsession &" to run the standard X11 session using vncserver. For the GNOME desktop, this line would read "gnome-session &."

The *-alwaysshared* option makes it possible for more than one user to simultaneously display the same desktop. The message tells the user that vncserver has created desktop 8 on host mira.

Next, the user and Bioadmin each type: **vncviewer mira:8** followed by the password, and the same vncviewer window will appear on both of their screens (*see* Fig. 2). If connecting via a browser, vncviewer would be launched by setting the URL to http://mira.cc.umanitoba.ca:5808, where the last two digits in 5808 indicate the screen number.

Now, both the user and Bioadmin can see and control the same desktop while discussing the various operations over the phone. The user can run a program that is causing difficulty, and the Bioadmin can see everything that happens. The Bioadmin can demonstrate in real time what the user should be doing, and if necessary, datafiles or configuration files such as .cshrc can be examined.

At the end of the session, the Bioadmin reminds the user to kill the VNC session by typing **vncserver -kill :8** and to change their VNC password.

The real value of VNC becomes apparent when traveling. For example, applications such as Powerpoint produce static presentations, and most people travel with their own laptop to ensure that it will work. Over the last 2 years, I have given real-time presentations across North America using VNC on any computer at hand. As long as a fast Internet connection is available, the full functionality of the desktop can be demonstrated anywhere there is a computer and a data projector.

Detecting, Handling and Preventing Problems

A multiuser system poses challenges in terms of managing shared resources, such as CPU time, memory, disk space, and network bandwidth. Usually it is possible to design a system that will minimize user errors, and in most cases UNIX is intrinsically protected from most catastrophes. For example, so that the permissions are explicitly set otherwise, a user can only read or modify files belonging to him, and usually these can only reside in the $HOME directory.

Disk Space

$HOME directories should always reside in a separate file system, and user quotas should be set, regardless of how much disk space exists. The one filesystem that is potentially troublesome is /tmp, which is writeable by all users. In the event that /tmp becomes full, programs that need to write temporary files may hang, resulting in a *filesystem full* error. The best way to avoid this problem is to have applications write temporary files to the current working directory, so that in the worst case, only the user is affected.

CPU Time

Monitoring CPU Usage

Keeping track of CPU usage is critically important. The **top** command gives you a real-time picture of the most CPU intensive jobs currently running on the host you are logged into. If you type **top** at the command line, your system will generate similar information to the following:

```
last pid: 13912; load averages: 2.61, 1.64, 1.31 13:48:41
504 processes: 488 sleeping, 1 running, 6 zombie, 7 stopped, 2 on cpu
CPU states: 16.4% idle, 65.7% user, 17.9% kernel, 0.0% iowait, 0.0% swap
Memory: 640M real, 17M free, 846M swap in use, 3407M swap free

  PID USERNAME THR PRI NICE  SIZE   RES STATE    TIME    CPU COMMAND
11371 umamyks   13  10    0   77M   71M cpu/0   23:19 64.27% matlab
27668 frist     10  58    0   97M   56M sleep    3:06  3.49% soffice.bin
13894 frist      1  33    0 3344K 1672K cpu/1    0:01  1.65% top
13898 umnorthv   1  58    0 6424K 4352K sleep    0:00  0.82% pine.exe
 1629 mills      7   0    0 9992K 7840K sleep    0:24  0.42% dtwm
13704 mhbasri    1  38    0 1464K 1360K sleep    0:01  0.31% elm.exe
 9797 syeung     1  58    0 1000K  816K sleep  267:53  0.28% newmail
 6914 umtirzit   8  58    0   13M 3992K sleep   26:38  0.23% dtmail
26524 mgarlich   1  58    0 9376K 6960K sleep    0:10  0.23% dtterm
29993 simosko    1  58    0 6824K 4528K sleep    0:21  0.23% pine.exe
 7937 jayasin    1  58    0 6112K 3816K sleep    6:55  0.22% Xvnc
 4483 francey    7  48    0 9904K 7920K sleep    0:24  0.21% dtwm
  206 root       6  58    0   76M 8024K sleep  458:13  0.20% automountd
27272 frist      7  58    0   11M 8376K sleep    0:35  0.20% dtwm
  580 syeung     1  48    0 2376K 1976K sleep    2:56  0.19% irc-2.6
```

This display is updated every few seconds in the terminal window. To quit, type **q**. The top command has many options. For example, you can sort jobs by memory used, or list only jobs under a given userid. The owner of a job can also kill that job using top. Type **man top** for full documentation.

The **ps** command with no arguments tells which jobs are running in the current shell (the current window):

```
ps
  PID TTY     TIME CMD
 2122 pts/104 0:00 dsdm
27401 pts/104 2:18 mozilla-
27376 pts/104 0:00 netscape
 2082 pts/104 0:00 zwgc
27384 pts/104 0:00 netscape
27396 pts/104 0:00 run-mozi
 2024 pts/104 0:18 Xvnc
 2041 pts/104 0:00 Xsession
27305 pts/104 0:01 csh
27381 pts/104 0:00 netscape
27457 pts/104 0:14 java_vm
```

while

ps -u userid

tells which jobs are running under a given userid on the host you are logged into.

The following list summarizes the types of tasks that tend to require a lot of processing time:

Jobs That Tend to be CPU-Intensive:

1. Phylogenetic Analysis
   a. Distance matrix methods (e.g., Neighbor Joining, FITCH): Very fast, the amount of time increases in a linear fashion with the number of sequences.
   b. Parsimony (e.g., DNAPARS, PROTPARS): Moderately efficient, the amount of time increases exponentially with the number of sequences
   c. Maximum likelihood (e.g., DNAML, PROTML, fastDNAML): Very slow, the amount of time increases according to a FACTORIAL function of the number of sequences.
2. Sequence database searches: The amount of time that is required is proportional to product of sequence length and database size; use high **k** values to speed up a search; protein searches are faster than DNA.
3. Multiple sequence alignments (e.g., CLUSTALX): Cluster type alignments scale linearly in proportion to the number of sequences.
4. Retrievals of large numbers of sequences: The time required is linear, related to number of sequences.
5. The efficiency of any sorting operation with a large number of items depends on the sort algorithm used.
6. Statistical and mathematical packages (e.g., SAS, MATLAB).

Jobs That Should Never Be CPU Intensive

If the following applications are using significant percentages of CPU time, they are not functioning normally, and are probably *runaway jobs*.

1. Graphic front ends: Programs such as GDE, SeqLab, or SeqPup by themselves do almost nothing. If you see one using a substantial amount of CPU time, it is probably a *runaway job*. One exception is when reading large sequence files, e.g., large numbers of sequences or very long sequences that are placed in memory for analysis.
2. Most user apps (e.g., word processors, mailers, spread sheets, drawing programs).
3. Desktop tools (e.g., text editors, filemanagers).
4. Most UNIX commands.
5. Web browsers: For short bursts browsers can be very CPU-intensive, but this should not persist for more than a minute or two.

Managing Long-Running Jobs

On any multitasking system, all jobs are assigned priorities that govern the amount of CPU time allocated to them. In UNIX, the nice command determines the priority. Most user commands default to a nice value of 0. This is especially important for applications run through a graphic interface, which need to work in real time. The higher the nice value, the less CPU time a job will be allocated, and the less of a load it puts on the system. Programs known to be CPU intensive can therefore be set to run at low priority. A higher nice value prevents the program from taking large amounts of time at high loads, but does not prevent it from utilizing CPU resources when the load on the system is light.

CPU-intensive tasks such as database searches or phylogenetic analysis should be run from wrappers, i.e., scripts that set parameters before running the program. The name of the program is preceded by the **nice** command. For example, to run fastDNAml at the default priority, a wrapper might contain the line:

```
nice fastDNAml arguments... &
```

The default priority for **nice** varies from system to system. In the BIRCH system, most sequence programs are launched from the GDE interface by calling wrappers that run programs using nice. As well, termination of the command line with an ampersand (&) tells the shell to run the task in the background. Thus, a user can launch a long-running job, quit GDE, and logout without terminating the job. When the program is completed, the output is written to the file, which the user can access when logged in during the next session.

In some cases, programs that use a graphic interface will perform analyses that require very long execution times. The problem with this design is that the user must remain logged in to the terminal from which the program was launched, because quitting the program would terminate the analysis. One can circumvent this problem by running jobs of this type from a vncviewer window. Killing a vncviewer window has no effect on the applications currently running, and the user can open up the same screen at any time from anywhere. This has the added benefit of making it easy to remotely monitor the progress of long-running jobs.

Killing Runaway Jobs

Sometimes a program will not correctly handle an error, and will begin using up large amounts of CPU time. Unless the Bioadmin has root permissions, it is necessary to ask either the owner of the job or a sysadmin to kill it. For example, a runaway netscape job might show up thus when running the top command:

```
PID    USERNAME  THR  PRI  NICE  SIZE  RES  STATE  TIME   CPU     COMMAND
25779  frist      13   22   0    24M   11M  cpu/0  23:19  64.27%  netscape
```

To kill the job, root or the owner would type: **kill -9 25779**.

Applications that do not normally use a lot of CPU time, but may be prone to runaway execution, could be contained by running them from a wrapper, in which the **ulimit** command is issued prior to running the program, e.g., **ulimit -t 900**, limiting CPU time in the current shell to 900 seconds (15 min).

Acknowledgments

Thanks to the Academic Computing and Networking staff at the University of Manitoba for UNIX system support. This work was made possible in part through hardware provided by the Sun Academic Equipment Grants Program.

Glossary and Abbreviations

ASCII Standard character set used by most computer operating systems, in which characters are represented by 8-bit numbers from 0 to 255.

PDF Portable document format. Platform-independent document rendering language for text and graphics. Files are more compact than PostScript. However, PDF viewers such as Adobe Acrobat typically generate PostScript, which is then printed. PDF was created by Adobe Systems Incorporated (*see* Website: www. adobe.com).

PostScript Standard control language used by virtually all laser printers for rendering text and graphics. Programs often create PostScript output which can be directly printed. PostScript was created by Adobe Systems Incorporated (*see* Website: www.adobe.com).

Suggested Readings

Introduction

Fristensky, B. (1999) Building a multiuser sequence analysis facility using freeware, in: Bioinformatics Methods and Protocols, (Misener, S. and Krawetz, S., eds.), Humana Press, Totowa, NJ, pp. 131–145.

Fristensky, B. (1999) Network computing: Restructuring how scientists use computers and what we get out of them, in: Bioinformatics Methods and Protocols, (Misener, S. and Krawetz, S., eds.), Humana Press, Totowa, NJ, pp. 401–412.

Sobell, M. G. (1995) A Practical Guide to the UNIX System, Benjamin/Cummings Publishing Co., Redwood City, CA.

Detecting, Handling and Preventing Problems

Felsenstein, J. (1989) PHYLIP Phylogeny Inference Package, Cladistics 5, 164–166.

Olsen, G. J., Matsuda, H., Hagstrom, R., and Overbeek, R. (1994) FastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood, Comp. Appl. Biosci. 10, 41–48.

Smith, S., Overbeek, R., Woese, C. R., Gilbert, W., and Gillevet, P. M. (1994) The Genetic Data Environment: an expandable GUI for multiple sequence analysis, Comp. Appl. Biosci. 10, 671–675.
(*see* Website: http://megasun.bch.umontreal.ca/pub/gde/)

Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F., and Higgins, D. G. (1997) The CLUSTAL X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools, Nucleic Acids Res. 25, 4876–4882.

C. Command Line Sequence Analysis

18 GCG File Management

Sittichoke Saisanit

Introduction

Users are most likely to encounter the Wisconsin Package (GCG) via the web interface as SeqWeb. As the name implies, SeqWeb is a web interface product that allows access to many programs in the GCG package. However, there are still a number of advantages for using GCG on the UNIX command line interface. First of all, the command-line interface is more amenable to batch processing of large datasets. Secondly, the command-line interface allows access to all programs not just the web interface subset. The use of GCG under the UNIX command line is presented in this chapter.

UNIX Commands and Overview

Familiarity with UNIXas presented in Chapter 13 and Appendix 3 is recommended prior to studying this chapter. Here are a few commands and rules in UNIX that can help you get started. Unlike DOS and VMS, UNIX is case-sensitive. For example, a file name *mygene.seq* is different from *Mygene.seq* or any other mixed case combinations. The **man** command is short for manual; it is equivalent to *help* in other operating systems and programs. For example, to find out how a certain UNIX command can be used, type **man** and the command of interest, then hit **Enter** at the command prompt %. For example: % **man cd**.

Manual pages for the command **cd** will be displayed. The **cd** command is used to change directory from one location to another. For example: % **cd /usr/home /usr/common/myproject**. This command changed the current working directory from /usr/home to /usr/common/myproject.

What if the command itself is not known? One powerful feature of the man pages is the ability to include a modifier *-k* to use a keyword feature for finding a command. For example, to find a command to delete a file, enter % **man -k remove**.

The command will list titles of man pages that contain the word *remove* in them. Introduction to UNIX is covered in Chapter 13 which includes descriptions of many UNIX commands.

In order to use GCG effectively with the command line interface, it is important to learn how to manipulate files and directories. This is fundamental to any operating system.

Using Database Sequences and Sequence Formats

The GCG package needs to be initialized by sourcing two scripts. This can be automated at the user log-in as the *.cshrc* file.

There are several formats of sequence data. Users may be familiar with GenBank or EMBL format. GCG has its own format. It also provides several program utilities to convert sequences from one format to another. The GCG format has one notable signature, i.e., 2 dots (..) to separate annotation and the sequence itself. The annotation proceeds the 2 dots followed by the sequence. In most cases, users should not worry about the GCG sequence format. All one needs to do is learn how to retrieve or specify a sequence from these databases when executing a GCG program. The GCG convention for specifying a sequence is *database:accession* or *database:locus_name*. The *database*, is the GCG logical name for a database. These names have been set by the GCG administrator. For example, it is customary to set **gb** for a logical name of GenBank database. To find out whether or not a local installation of GCG has **gb** as one of the logical names, issue the following command: % **name gb**.

To list the logical names, issue the command name without any database name.

Example: % **name**

To retrieve a sequence, the GCG **Fetch** command can be used.

Example: % **fetch gb:m97796**

Assuming that the GenBank database is installed locally and *gb* is set as its logical name, the above command will retrieve a GenBank sequence which has an accession number of M97796. The result will be written to a file with a default name unless a name is specifically given to the program.

The *Fetch* program can also work without database name specification.

Example: % **fetch m97796**

However, if the accession number appears in more than one database, *Fetch* will retrieve all of the sequence records. To ensure uniqueness and speed of retrieval, it is best to use *Fetch* with full specification of the database name and sequence accession number.

On the occasion that a GCG sequence is created or modified by a text editor and the checksum has been altered, GCG programs will not recognize this sequence. Users need to run a utility called *reformat* (shown below) to correct the checksum.

Example: % **reformat myseq.seq**

Another useful file format in GCG is the Rich Sequence Format (RSF). In SeqLab, which is a graphical user interface for GCG run under X Windows, RSF is particularly useful because sequence annotations such as domains and phosphorylation sites can be displayed for visualization. SeqLab can only be run from a UNIX workstation or an X Windows emulation program. There are two modes of working when inside SeqLab: *main list* and *editor*. The graphical sequence viewer is available in SeqLab *editor* mode. The **reformat** command can be used to convert a GCG sequence into an RSF format sequence by including the **-RSF** parameter in the command line: % **reformat -rsf myseq.seq**.

It is recommended that users name sequence files consistently. By default, GCG does not require consistent naming and UNIX does not insist on file types. In contrast, DOS and Windows usually require a 3-letter file type extension for files to be recognized correctly by the programs. However, after accumulating files it will be difficult to recognize older files. Users should make a habit of naming sequence files with meaningful names and consistent name extensions. Appending *.dna* or *.seq* extension for nucleic acid sequence files, or *.pep* or *.pro* extension for protein sequence files will help in recognizing these files. In addition, storing sequence files in specific directories for each project is generally a good idea. Once a certain project has been completed, the entire directory can be archived or removed.

Another file format users may encounter is the FASTA format. Most public sequence utilities on the web can accept or produce sequences in FASTA format. GCG has a utility to convert GCG sequences into FASTA format sequences and vice versa. This is useful because it allows one to use other available tools and external sequences. **Tofasta** converts a GCG sequence into a FASTA format sequence.

Example: % **tofasta gb:m97796**

FromFasta converts a FASTA sequence into a GCG format sequence.

Example: % **fromfasta pubseq.fasta**

Editing GCG Formatted Sequences

The need to edit sequences may come from users' own sequencing efforts. Additionally, users may want to track recombinant sequences such as products of mutagenesis. **SeqEd** is a utility to edit sequences in a much more efficient manner than a text editor. **SeqEd** has another advantage in that the edited sequence will be recognized by other GCG programs without the need to run the **reformat** command. Annotations to specific residues can also be placed within an edited sequence. **SeqEd** can be started by entering: % **seqed myseq.seq**.

Once inside a **SeqEd** editor, use **Control-D** to enter the editor command. Entering **help** in the editor brings up a list of commands that can be used inside the editor.

List File

When working with a family of gene or protein sequences, there is often a need to simultaneously manage a number of sequences. GCG provides a powerful function called a *list file*. A list file is simply a text file that contains a list of individual sequences beginning with 2 dots (..) and separated by new lines. The GCG programs ignore any text before the 2 dots and any text after an exclamation mark (!). Therefore, comments or descriptions of sequences can be added. An example of a list file:

```
Sequences of mammalian EGF receptors and related family members.
..
sw:EGFR_HUMAN
sw:EGFR_MOUSE
/usr/home/newdata/myseq.pep ! unpublished EGFR-related sequence
```

As shown in this example, a list file can contain either database sequences or local user sequences or both. A list file is accessed by preceding the file name with the @ symbol. For example, to retrieve all sequences in the list file named *egfr.list* to the current working directory, use the GCG **Fetch** command:

% **fetch @egfr.list**.

In addition to making a list file, multiple sequences can be aligned and written to a single Multiple Sequence Format (MSF) file. Several GCG programs can output files in an MSF format. For example, the **reformat** program with the **-MSF** parameter can be used to convert a group of sequences from a list file into an MSF formatted file.

Example: % **reformat -msf @egfr.list**

However, **reformat** does not align the input sequences. The file resulting from reformat can be named *egfr.msf*, for example. This MSF file can then be used as input for other GCG programs. One or a subset or all of the sequences in an MSF file can be used. To specify a single sequence from an MSF file, type the MSF file name followed by the sequence name in curly brackets, for example **egfr.msf{egfr1}**. To specify multiple sequences, an asterisk wildcard character must be used. For example, **egfr.msf{egfr*}** specifies sequences in the *egfr.msf* file with sequence names beginning with *egfr*. Similiary, **egfr.msf{*}** indicates that all sequences in the MSF file will be used. Note, plain file name specification is not sufficient to specify sequences from MSF files. Either a sequence name or wildcard in the curly brackets must be used with the file name.

SeqLab, the X Windows interface for GCG, can also output MSF files from a list of sequences. GCG command-line programs that can output MSF files are listed below. Programs that require **-MSF** parameter are listed accordingly.

- LineUp -MSF
- PileUp
- PrettyBox
- ProfileGap -MSF
- ProfileSegments -MSF
- Reformat -MSF

Below are two examples of how to use MSF files in a program without (PileUp) and with (LineUp) "-MSF" option requirement.

Example: % **pileup egfr.msf{*}**
% **lineup -msf egfr.msf**

Graphic Files

Several GCG programs have an option to generate output in a graphic format. In order to use the graphic feature, a graphical language and a graphic device must be defined. The command **ShowPlot** displays the current graphic device while the command **SetPlot** changes it. After setting the graphic device, the command **PlotTest** can generate a test graphic output. It is a quick and easy way to determine whether the device is properly configured.

Graphic files require specific applications in order to be displayed correctly. They can not be displayed from the command-line interface like plain text files. The *.figure* files are generally a graphic output from many GCG programs.

Graphics can be displayed directly on the screen. If an appropriate device is selected. For example, on an X Windows terminal, *ColorX* can be used. *ColorX* is a graphic language and a device for the X Windows environment.

File management in GCG requires knowledge of the operating system on which GCG runs. Most likely, it is one of many flavors of UNIX. Common sense should be applied to maintain naming consistency and to facilitate the task of file organization. This is helped by the various file utilities for creating sequence files and converting them into proper formats. Learning how to manage and use graphic files will be helpful to visualize the output from many GCG programs.

Glossary and Abbreviations

EMBL Nucleotide Database Europe's primary collection of all publicly available nucleotide sequences. It is maintain in collaboration with GenBank and DDBJ (Japan).

GCG Genetics Computer Group started in 1982 within the Department of Genetics at the University of Wisconsin. It went private in 1990 and was acquired by Oxford Molecular Group in 1997. In 2000, Oxford Molecular was acquired by Pharmacopeia resulting in a new company called Accelrys (*see* Website: http://www.accelrys.com) which is currently the commercial distributor of GCG(r) Wisconsin Package™.

GenBank An annotated collection of all publicly available nucleotide sequences. The protein sequence collection is referred to specifically as GenPept. GenBank is maintained by National Center of Biotechnology Information (NCBI), a unit of the US National Institute of Health (NIH).

SWISS-PROT An annotated protein sequence database maintained and curated by the Swiss Institute for Bioinformatics (SIB). The database designation is often abbreviated as SW in GCG.

Wisconsin Package A suite of tools and programs for Bioinformatics sequence analysis developed by GCG. It runs on various UNIX operating systems including SUN Solaris, SGI IRIX, Compaq Tru64 UNIX, IBM AIX, and Red Hat Linux.

19 GCG Sequence Analysis

Sittichoke Saisanit

Introduction

The strength of the Wisconsin Package (GCG) is that it contains a suite of programs that serve a wide range of sequence analysis needs. A selected number of these programs are described. Each GCG program has extensive modifiers or parameters available. Top-level online help contains a list of all programs and appendices in alphabetical order. Lists of IUPAC standard symbols for amino acids and nucleotides, including ambiguous symbols, can be found in Appendix 2. Program manuals organize the GCG programs into groups of related analyses or functions. The user's guide contains introductions and overviews that provide an excellent resource for those who are new to GCG. Release notes provide a list of new programs and existing programs with enhancements or bug fixes. To access top-level online help, program manuals, the user's guide and release notes, simply type **genhelp** without specifying a program name after the command prompt % (**genhelp**).

For specific help, i.e, to get help on the Map program, enter: % **genhelp map**.

Individual Sequence Analysis

Several programs in GCG can be used for analysis of individual DNA or protein sequences. These programs are divided into the functional groups of DNA restriction mapping, pattern recognition, secondary structure prediction, RNA secondary structure prediction, and other protein analysis.

DNA Restriction Mapping

Restriction digestions are routine laboratory techniques that many molecular biologists perform. There are related pattern-recognition programs in GCG that can facilitate the analysis or the preparation of a restriction digestion experiment. They are *Map*, *MapSort*, and *MapPlot*.

Map locates restriction sites within a given sequence. It also provides an optional protein translation along with the result. For example, upon entering the **Map** command, the program recognizes the sequence specification convention *database: accession number* (*gb:m97796*), and immediately uses the database sequence as an input. It then asks the user to enter specific restriction sites. All sites can be selected by hitting **return** key or none by hitting the **space bar**, followed by the **return** key. Next, the program asks the user to specify which frame is to be used for protein translation. Select a specific frame if known. If not known, select 3 forward frames, 3 reverse frames, or open reading frames. Otherwise, select the none option in order not to view any protein translation. Map then generates an output text file, which can be viewed by using the UNIX command **more** or by any other text viewer/editor.

All prompted parameters can also be entered in a single line of command. Following the interactive example earlier, the **map** command with needed parameters can be entered with the required parameters.

Example: % **map gb:m97796 -enzyme=* -menu=o -out=m97796.map -default**

The **-enzyme=*** parameter means that all restriction sites (*) are chosen. With the **-menu=o**, protein translation is performed for open reading frames (**o**) only. To save an output into a file named *m97796.map*, the parameter **-out=m97796.map** is supplied. Finally, the **-default** parameter tells the Map program to accept default values for any prompted parameter not specified. In this case, the start and the end positions of the sequence for the Map program to run are not specified. Therefore using the **-default** parameter, the entire sequence is analyzed.

Like the Map program example, any GCG command along with the necessary parameters can be submitted in a single line. This is very useful when using a GCG command inside a UNIX shell script. Note that it is a good idea to always append the *-default* option to any command in order to make sure that a default value is selected for only the parameter that was not specified. Running GCG in a shell script enables complex possibilities such as a multi-step analysis and repetitive or regularly scheduled processes.

Another related program for restriction mapping is MapSort. MapSort finds the restriction enzyme cuts in a given DNA sequence and sorts the fragments of the resulting theoretical digestion by size. MapSort can be used to predict how the fragments of a restriction enzyme reaction will separate on a gel. It is possible to concatenate a sequence of interest with its vector before running MapSort to determine if a single step isolation is possible. With the optional parameter **-circular**, the sequence will be analyzed in its circular form, otherwise the program considers the sequence as linear by default.

Example: % **mapsort myrecomb.seq -circular**

Lastly, the MapPlot program graphically displays restriction sites. In addition to a graphic output, MapPlot can write a text file that also approximates the graph. The program helps users visualize how the part of a DNA molecule of interest may be isolated.

Finding rare-cutter restriction enzymes in a given sequence is often a useful exercise. These enzymes can then be used to quickly test whether a clone or a polymerase chain reaction (PCR) product is likely to be correct. For example, if a *Eco*RI digestion of a 600 bp PCR product is predicted to produce 2 fragments of 400 and 200 bp each, it is prudent to digest a portion of the PCR product and examine it along side the undigested material using electrophoresis. Observing all three different size species will give confidence that the PCR product obtained is not an artifact. One optional parameter for mapping programs is **-six**. This option causes a program like Map to find only enzymes that have six base recognition sites (rare cutters). For subcloning experiments, it is often desirable to identify enzymes that cut only once in a sequence. Option **-once** is used to this end.

When using a mapping program, GCG provides a standard restriction enzyme site file (enzyme.dat) from the New England Biolabs for all commercially available enzymes. Single representatives of isoschizomers are defined in the file. However, there is an option to create a user-customized file. For example, it may be useful to run the Map program against a set of enzymes that are only available in the laboratory. In addition, there may be a preference for different isoschizomers not selected in the default *enzyme.dat* file. One way to create this customized file is to **Fetch** the **enzyme.dat** file and rename it to a personalized name, e.g., **labenzyme.dat**.

```
Example: % fetch enzyme.dat
         % mv enzyme.dat labenzyme.dat
```

Starting with a list of all availble enzymes, the file **labenzyme.dat** can be modified to exclude enzymes not available in the laboratory. Inserting exclamation marks (**!**) in front of these enzymes will cause the mapping program to ignore them. In addition, preferred isoschizomers that were not selected as default can be chosen by removing the corresponding exclamation marks. To run the map program with this local data file, use option **-data** command to specify the file with:

```
% map gb:m97796 -data=labenzyme.dat
```

Pattern Recognition

Map and related programs are essentially pattern recognition programs. GCG also has a general purpose pattern finding program called *FindPatterns*. FindPatterns identifies sequences that contain short patterns like CANNTG or GCAATTGC. Ambiguous patterns can also be used. FindPatterns can recognize patterns with some symbols mismatched but not with gaps. For example, the optional parameter **-mis=1** allows 1 mismatch to occur in the search pattern. Patterns can be provided in a file or typed during an interactive session. It supports the IUPAC nucleotide code.

```
Example: % findpatterns myseq.seq -data=pattern.dat
```

The GCG formatted search patterns can be provided as a local file. Alternatively, the patterns can be individually entered from the command line. If using the command line to enter patterns, simply hit **return** when the last pattern has been entered and after the program has again prompted for the next pattern to be entered.

Output from FindPatterns is a text file that can be viewed by any text viewer/editor. This result shows only those sequences in which a pattern was found and its location. Use of the **-show** parameter will include a list of patterns not found.

FindPatterns works with either nucleotide or protein sequences. To search through proteins for sequence motifs as defined in the PROSITE Dictionary of Proteins Sites and Patterns, the program *Motifs* should be used. Motifs also displays a short literature summary for each of the motifs it finds in a given sequence. The program can recognize patterns with some of the symbols mismatched, but not with gaps.

Other pattern finding utilities can be quite useful when special analysis needs arise. The Map program can display open reading frames for the six possible translation frames of a DNA sequence. However, the program *Frames* was created specifically to find open reading frames. The program *Repeat* finds direct repeats in a sequence. A minimum repeat length (window), a stringency within the window, and a search range are required. Repeat then finds all the repeats of at least that size and stringency within the search range. The *Composition* program determines the composition of either DNA or protein sequences. To calculate a consensus sequence for a set of prealigned nucleic acid sequences, the program *Consensus* can be used. *Consensus* tabulates the percent of G, A, T, and C for each position in the alignment. *FitConsensus* can then use the output table as a query to search for patterns of the derived consensus in other DNA sequences.

Secondary Structure Prediction

Prediction of protein secondary structures is another suite of programs that is used quite frequently. The *PeptideStructure* program makes secondary structure predictions of an amino acid sequence. PeptideStructure calculates several predictions using the following methods:

- Secondary structure according to the Chou-Fasman method
- Secondary structure according to the Garnier-Osguthorpe-Robson method
- Hydrophilicity according to either the Kyte-Doolittle or Hopp-Woods method
- Surface probability according to the Emini method
- Flexibility according to the Karplus-Schulz method
- Glycosylation sites
- Antigenic index according to the Jameson-Wolf method

The results of PeptideStructure are written into a file for graphical presentation using a program called *PlotStructure*. The predicted structures can be shown with two main options: 1) parallel panels of a graph, or 2) a two-dimensional squiggly representation. With the first option, PlotStructure creates a one-dimensional, multi-paneled plot, in which the residues are numbered on the x-axis, and the attributes are represented as continuous curves in each of several different panels. The horizontal line across the surface probability panel at position 1.0 on the y-axis indicates the expected surface probability calculated for a random sequence. Values above this line indicate an increased probability of being found on the protein surface.

The *squiggly plot* option generates a two-dimensional representation of predicted secondary structures with different wave forms. Helices are shown with a sine wave, beta-sheets with a sharp saw-tooth wave, turns with 180 degree turns, and coils with a dull saw-tooth wave. Any of four different quantitative attributes (hydrophilicity, surface probability, flexibility, or antigenic index) can be superimposed over the wave with special symbols wherever the attribute exceeds a set threshold. The size of the symbols is proportional to the value of the attribute. In addition, possible glycosylation sites can be marked on the two-dimensional plot.

As with any computational prediction, users should be cautioned that they are just predictions and should be treated as such. They are not substitutes for experimental proof but can help to direct or focus experiments for validation.

Another program that determines peptide secondary structure is *PepPlot*. PepPlot calculates and plots parallel curves of standard measures for protein secondary structure. Most of the curves are the average, sum, or product of some residue-specific attribute within a window. In a few cases, the attribute is both specific to the residue and dependent on its position in the window. Throughout the plot, the blue curves represent beta-sheets and the red curves represent alpha-helices. Black indicates turns and hydropathy.

There are 10 different panels that can be plotted in any combination and in any order. The first part of the plot shows the sequence itself. This panel is extremely crowded if more than 100 residues per page are selected. When using PepPlot, consult the online help.

Other Protein Analysis

Numerous other programs can be used to analyze proteins. Short descriptions of each are given below. You should consult the online help for detailed descriptions and to find optional parameters for that analysis.

- *SPScan* can be used to predict secretory signal peptides (SPs) in protein sequences. For each sequence, SPScan prints a list of possible secretory signal peptides sorted by score.
- *Isoelectric* calculates the isoelectric point of a protein from its amino acid composition. Isoelectric plots the positive and negative charges and the charge of a protein as a function of pH.
- Hydrophobic *moment* is the hydrophobicity of a peptide measured for different angles of rotation per residue. The hydrophobic moment is calculated for all angles of rotation from 0–180 degrees.
- *HelicalWheel* plots a helical wheel representation of a peptide sequence. Each residue is offset from the preceding one by 100 degrees, the typical angle of rotation for an alpha-helix.
- Like *Map* for a nucleotide sequence, *PeptideMap* finds positions where known proteolytic enzyme or reagent might cut a peptide sequence.
- *PeptideSort* cuts a peptide sequence with any or all of the proteolytic enzymes and reagents listed in the public or local data file *proenzall.dat*. The peptides from each digestion are sorted by position, weight, and retention time as if subjected to high-performance liquid chromatography (HPLC) analysis at pH 2.1.

RNA Secondary Structure Prediction

Designing an anti-sense oligonucleotide is aided by folding predictions that would indicate single-stranded regions. There are 2 GCG programs that can be used. First, *StemLoop* searches for stems or inverted repeats in a sequence. However, StemLoop cannot recognize a structure with gaps, i.e., bulge loops or uneven bifurcation loops. The stems can be plotted with the *DotPlot* utility. Second, *MFold* predicts optimal and suboptimal secondary structures for an RNA or DNA molecule using the most recent energy minimization method of Zuker. PlotFold then displays the secondary structures predicted by Mfold.

Pairwise Analysis

Comparison of two sequences is undertaken to identify related segments. *Gap* and *Bestfit* are two GCG programs that can be used to compare 2 sequences of the same kind, i.e., two DNA sequences or two protein sequences. Gap creates a global alignment of the entire input, while *Bestfit*, as the name suggests, locally aligns the most similiar regions.

Gap uses the Needleman and Wunsch algorithm to create a global alignment of two sequences maximizing the number of matches and minimizing the number and size of gaps. Matching residues receive scoring value assignments. In addition, a *gap creation penalty* and a *gap extension penalty* are required to limit the insertion of gaps in the alignment. These parameters can be adjusted for best results. For example, the gap creation penalty and the gap extension penalty can be set low when two sequences are known to be only related in major domains, but to differ elsewhere. In this case, a gap between the domains is expected and therefore should not be significantly penalized. However, too low a penalty could allow too many gaps in the alignment. Depending on each case, several adjustments of the gap creation and gap extension penalties may be required before the best alignment is obtained.

Bestfit finds an optimal alignment of the best segment of similarity between two sequences using the Smith and Waterman algorithm. Optimal alignments are found by inserting gaps to maximize the number of matches. The two sequences can be of very different lengths and have only a small segment of similarity. If the relationship of two sequences is unknown, Bestfit is the most powerful method in the GCG Package for identifying the best region of similarity between sequences.

Gap and Bestfit normally produce output in plain text format. To display an alignment in a graphical format that shows the distribution of similarities and gaps, *GapShow* can be used. The two input sequences should be aligned with either Gap or Bestfit before they are given to GapShow for display. Gap and Bestfit can write the aligned sequences (with gaps inserted) into new sequence files. GapShow reads these files and plots the distribution of the differences or similarities in the alignment. The sequences are represented by horizontal lines. These horizontal lines have openings at points where there are gaps in either sequence. Regions of interest, such as coding regions, can be shown outside these lines. With appropriate options, a large vertical line between the sequences indicates either a difference or similarity.

FrameAlign can align a protein and the coding sequence, corresponding to the mRNA. FrameAlign creates an optimal alignment of the best segment of similarity (local alignment) between the protein sequence and the codons in all possible reading frames on a single strand of a nucleotide sequence. FrameAlign inserts gaps to obtain the optimal local alignment. Because FrameAlign can align the protein to codons in different reading frames of the nucleotide sequence, it can identify sequence similarity even when the nucleotide sequence contains reading frame shifts.

In standard sequence alignment programs, gap creation and extension penalties must be specified. In addition to these penalties, FrameAlign also has an option to specify a separate frameshift penalty for the creation of gaps that result in reading frame shifts in the nucleotide sequence. By default, FrameAlign creates a local alignment between the nucleotide and protein sequences. With the **-global** parameter, FrameAlign creates a global alignment where gaps are inserted to optimize the alignment between the entire nucleotide sequence and the entire protein sequence.

Developing a habit of consulting the online help prior to using any of these programs is a good idea. It will enable a full utilization of the programs and all of their options and flexibility.

Glossary and Abbreviations

Isoschizomers Different restriction enzymes that recognize and cut the same nucleotide sequence pattern.

PROSITE A database of protein families and domains. It consists of biologically significant sites, patterns and profiles that help to reliably identify to which known protein family (if any) a new sequence.

Suggested Readings

Individual Sequence Analysis

Zuker, M. (1989) On finding all suboptimal foldings of an RNA molecule, Science 244, 48–52.

Pairwise Analysis

Needleman, S. B. and Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48, 443–453.

Part IV Computer Applications

A. Management and Analysis of DNA Sequencing Projects and Sequences

20 Managing Sequencing Projects in the GAP4 Environment

Rodger Staden, David P. Judge, and James K. Bonfield

Introduction

Methods for managing large scale sequencing projects are available through the use of our GAP4 package and the applications to which it can link are described. This main assembly and editing program, also provides a graphical user interface to the assembly engines: CAP3, FAKII, and PHRAP. Because of the diversity of working practices in the large number of laboratories where the package is used, these methods are very flexible and are readily tailored to suit local needs. For example, the Sanger Centre in the UK and the Whitehead Institute in the United States have both made major contributions to the human genome project using the package in different ways. The manual for the current (2001.0) version of the package is over 500 pages when printed, so this chapter is a brief overview of some of its most important components. We have tried to show a logical route through the methods in the package: pre-processing, assembly, contig[1] ordering using read-pairs, contig joining using sequence comparison, assembly checking, automated experiment suggestions for extending contigs and solving problems, and ending with editing and consensus file generation. Before this overview, two important aspects of the package are outlined: the file formats used, the displays and the powerful user interface of GAP4. The package runs on UNIX and Microsoft Windows platforms and is entirely free to academic users, and can be downloaded from Website: http://www.mrc-lmb.cam.ac.uk/pubseq.

The Data and Its Accuracy

The aim of a sequencing project is to produce a final contiguous sequence that satisfies certain accuracy criteria. During the project, the software should show which sequence positions fail to meet the accuracy standard and enable them to be dealt with efficiently. For us, the key to minimizing user interaction with sequence assembly data lies in the use of consensus algorithms, which employ base accuracy estimates or confidence values. Within GAP4 these algorithms are used when automatically locating problems that require an experimental solution and also during interactive tasks within the *Contig Editor* where they are used to guide the user only to those positions

[1] There is currently some confusion about the usage of the word *contig*. In this chapter the original meaning is used, a *contig* is defined to be a set of overlapping segments of DNA, Staden, R. (1980).

whose consensus accuracy falls below a given standard, and hence that require human attention. The consensus is updated on the fly for each change to the data. Because the same algorithm is used during editing and to create the final sequence, the user can ignore all apparent conflicts in the aligned readings, other than those that fail to produce A,C,G, or T in the consensus. Moreover, if the consensus at any point is a padding character (in GAP4 denoted by " * "), these too can be ignored as they will be removed from any consensus sequence file calculated by the program. The sole role of pads is to permit alignment of readings without the need to delete the original bases.

File Formats: SCF, ZTR, Experiment, Fasta Files, and the GAP4 Database

Suitable file formats for sequence related data can make it easier to write programs to process the data and can save a great deal of disk space. For storing traces, sequences and accuracy values, for data from fluorescence-based sequencing machines, we invented the SCF format. Later we moved to SCF version 3.0, which contains the same information as earlier versions but in a form that can be easily compressed, achieving a compression ration of around 10:1 using gzip on data converted from ABI files. Our programs automatically uncompress these files on the fly. More recently we have created the ZTR format, which produces even smaller files without the need for external compression programs. Experiment files are used for passing sequence readings and related data between processing programs. A list of some of the experiment file record types is shown in Fig. 1. Experiment files and FASTA format files can be used for output of the consensus sequence. The GAP4 program uses experiment files to import reading data for assembly and stores it in its own database. In addition to storing the items marked with asterisks (*) in Fig. 1, GAP4 records the positions of all the edits made to individual readings, and provides methods to check their validity. The database is designed so that in the event of system failure, although the most recent changes to it may be lost, it should not get corrupted.

Pre-assembly

Prior to assembly into a GAP4 database, sequence readings produced by automated sequencing instruments must be passed through several processes. Typically this will include conversion to SCF/ZTR format, calculation of base calling accuracy or confidence values, quality clipping, sequencing vector clipping, cloning (e.g., BAC) vector removal, and repeat tagging. A comprehensive set of operations is shown in the flowchart of Fig. 2. Using our package, each of these individual steps is performed by a separate and specific program. These routine operations, which generally require no user intervention, are configured and controlled by the program PREGAP4, which can pass any number of readings through each of these processes in a single run.

The input to PREGAP4 is a file containing the names of all the sequencing instrument files to process. The output is generally an SCF/ZTR file and an Experiment file for each of the input files processed, plus a new file of file names containing the names of all the newly created Experiment files, ready to be passed to GAP4. That is, PREGAP4 creates the initial Experiment file for each reading and then sends it, in turn, through each of the required processing steps. PREGAP4 is modular and very flexible, and can be tailored for compatibility with local working practices. It can be configured to work completely automatically, or to be partially interactive.

AC	Accession number
AP	Assembly Position
AQ	Average Quality for bases 100..200
AV	Accuracy values for externally assembled data **
BC	Base Calling software
CC	Comment line
CF	Cloning vector sequence File
CL	Cloning vector Left end
CN	Clone Name
CR	Cloning vector Right end
CS	Cloning vector Sequence present in sequence *
CV	Cloning Vector type
DR	Direction of Read
DT	DaTe of experiment
EN	Experiment Name
EX	EXperimental notes
ID	IDentifier *
LE	was Library Entry, but now identifies a well in a micro titre dish
LI	was subclone LIbrary but now identifies a micro titre dish
LN	Local format trace file Name *
LT	Local format trace file Type *
MC	MaChine on which experiment ran
MN	Machine generated trace file Name
MT	Machine generated trace file Type
ON	Original base Numbers (positions) for externally assembled data **
OP	OPerator
PC	Position in Contig for externally assembled data **
PN	Primer Name
PR	PRimer type *
PS	Processing Status
QL	poor Quality sequence present at Left (5') end *
QR	poor Quality sequence present at Right (3') end *
SC	Sequencing vector Cloning site
SE	SEnse (ie whether complemented) **
SF	Sequencing vector sequence File
SI	Sequencing vector Insertion length *
SL	Sequencing vector sequence present at Left (5') end *
SP	Sequencing vector Primer site (relative to cloning site)
SQ	SeQuence *
SR	Sequencing vector sequence present at Right (3') end *
SS	Screening Sequencing
ST	STrands *
SV	Sequencing Vector type *
TG	Gel reading Tag
TC	Consensus Tag *
TN	Template Name *
//	End of sequence records (SQ)

Fig. 1. Experiment file record types.

A screen dump from PREGAP4 is shown in Fig. 3. The program menus are at the top and the main window is occupied by the *Configure Modules* page. This page allows users to select and configure the modules they require: those selected are marked with a tick, the others with a cross. If a module is fully configured, it is labeled *ok*. Here the user is setting up the module that is used to locate and mark the sequencing vector. When everything is ready, the **Run** button at the bottom can be used to activate the processing.

Introduction to GAP4

GAP4 provides a comprehensive set of methods for: assembly; checking assemblies; finding joins between contigs by using read-pair data and/or poor quality data at

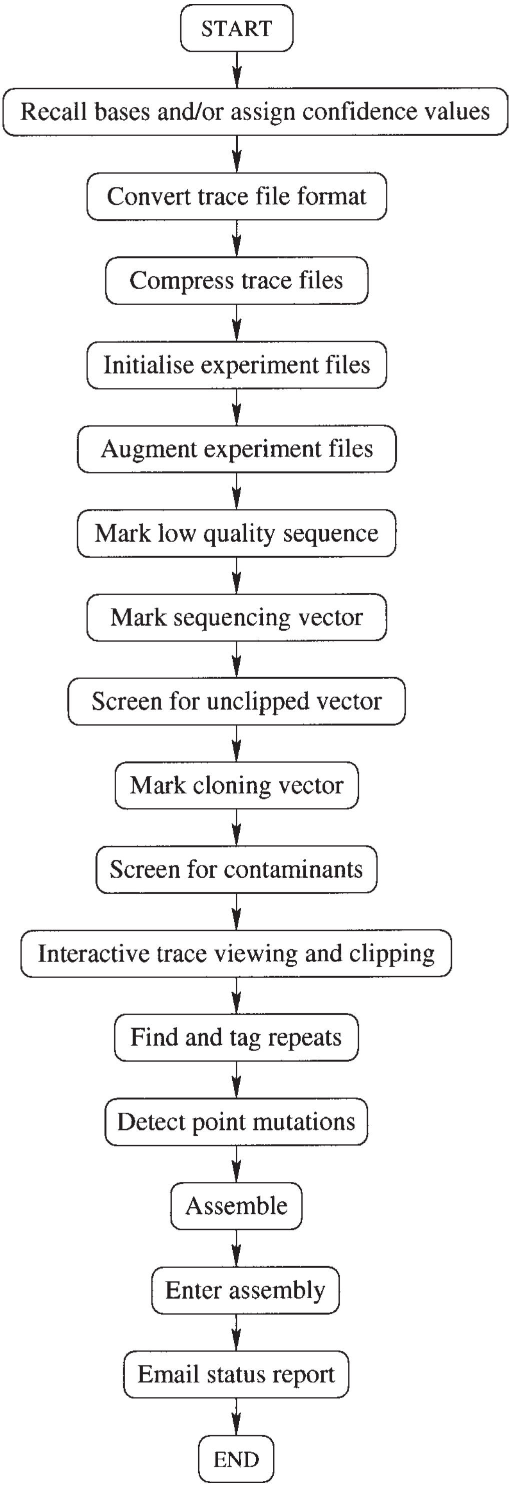
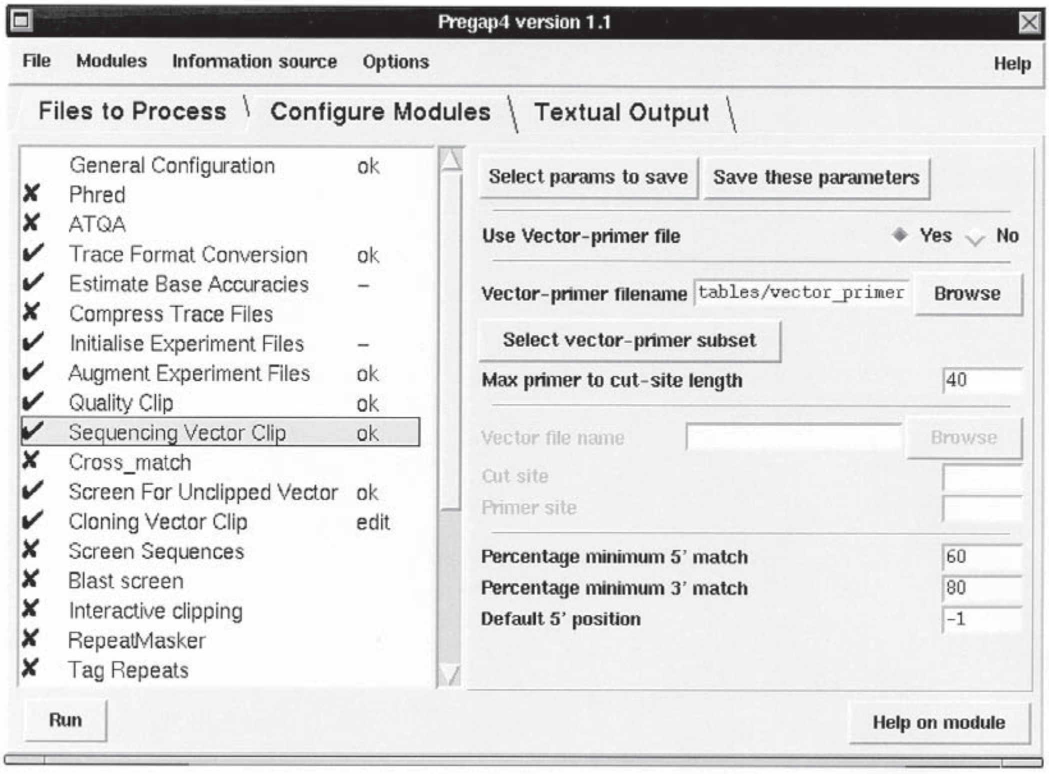


Fig. 2. A flowchart of the operations configured and carried out by PREGAP4.



	General Configuration	ok
✗	Phred	
✗	ATQA	
✓	Trace Format Conversion	ok
✓	Estimate Base Accuracies	–
✗	Compress Trace Files	
✓	Initialise Experiment Files	–
✓	Augment Experiment Files	ok
✓	Quality Clip	ok
✓	Sequencing Vector Clip	ok
✗	Cross_match	
✓	Screen For Unclipped Vector	ok
✓	Cloning Vector Clip	edit
✗	Screen Sequences	
✗	Blast screen	
✗	Interactive clipping	
✗	RepeatMasker	
✗	Tag Repeats	

Use Vector-primer file ● Yes ○ No

Vector-primer filename tables/vector_primer

Max primer to cut-site length 40

Vector file name

Cut site

Primer site

Percentage minimum 5' match 60

Percentage minimum 3' match 80

Default 5' position -1

Fig. 3. The main PREGAP4 window.

the contig ends; suggesting additional specific sequencing experiments to join contigs or to overcome other deficiencies in the data; check the accuracy of the consensus; and edit contigs to the required level of confidence.

Files of File Names, Lists, Tags, Masking, Visible and Hidden Data

Files of file names are files that contain the names of other files that are to be processed as a batch. As already mentioned, PREGAP4 processes batches of files from sequencing instruments and the various GAP4 assembly engines process batches of experiment files. Both do this by being given the name of a file of file names. Lists are used internally by GAP4 for recording sets of readings or contigs, and to enable them to be processed in batches. Tags are labels that can be attached to segments of individual readings or consensus sequences. Each tag has an associated, editable text string and each tag type has an identifier and color. Tags can be used to record comments about segments of a sequence, or to denote the locations of specific features, such as primers or repeat elements. Within GAP4 tags can be displayed at all levels of resolution to provide visual clues, and can be used to mark segments of the sequence for special treatment by certain algorithms. For example, segments covered by selected tags can be *masked* during assembly. Our pre-processing programs do not actually remove vector sequences or poor quality data from the ends of readings, rather they add records to the experiment files to enable those segments to be treated in special ways within other programs such as GAP4. These end segments of readings are not normally revealed in the *Contig Editor* and other such places, and so they have come to be known as *hidden* data and the used parts of the readings as *visible*.

The GAP4 Scripting Language

GAP4 is composed of algorithms written in the C language, which have been added to the commands **understood** by the Tcl interpreter. Extended Tcl provides a scripting language for performing operations on sequence assembly data. For example, a simple script could open a GAP4 database, make a copy of it, and then assemble a new batch of readings listed in a file of file names. The design of GAP4 also makes it possible for others to add their own interactive modules, complete with dialogues and menu items.

The GAP4 User Interface

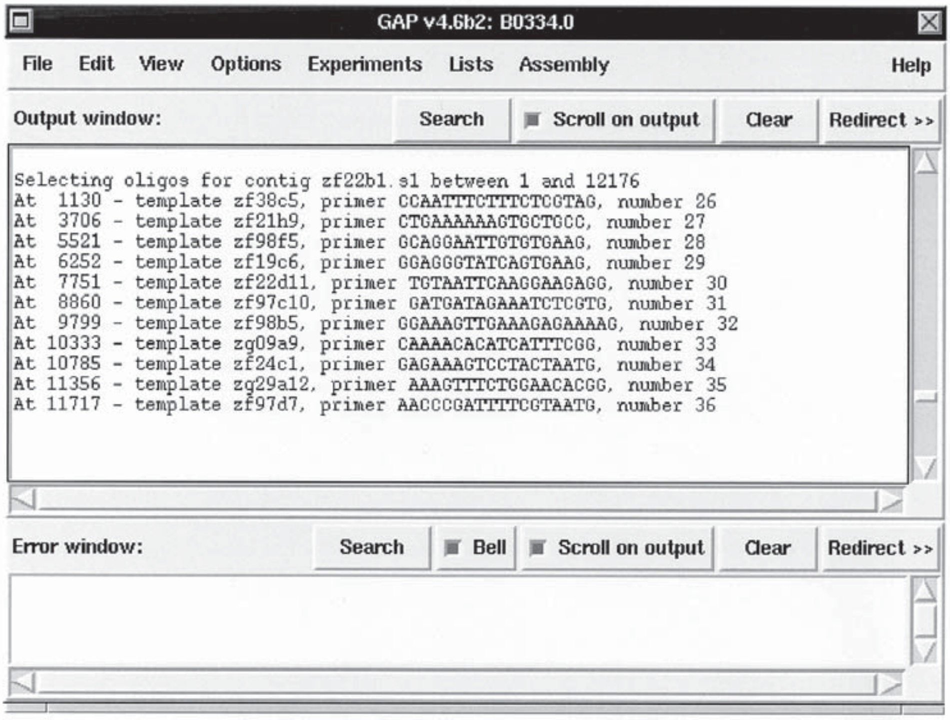
The graphical displays and user interface of GAP4 play an important role in enabling users to deal with difficult problems, and so help to simplify and speed up sequencing projects. The main points are summarised in the following sections.

The Main Window and Its Menus

An example of the main window of GAP4 is shown in Fig. 4. It consists of an area for receiving textual results, the *Output window*, and below that an area for displaying error messages. The output window is a searchable and has an editable text display, the contents of which can be saved to disk files. Along the top of the window is a row of menus: the file menu includes database opening and copying functions and consensus sequence file creation routines. The *Edit* menu contains options to alter the contents of the database such as **Edit Contig**, **Order Contigs**, **Join Contigs**, **Break Contig**, **Disassemble Readings**, **Complement Contig**, etc. The *View* menu provides functions that give different ways of examining the data that includes: **Contig Selector**, **Results Manager**, **Find Internal Joins**, **Find Read Pairs**, **Find Repeats**, **Check Assembly**, **Show Templates**, **Restriction Enzyme Map**, **Stop Codon Map**, **Quality Plot**, etc. The *Options* menu contains general configuration commands. The *Experiments* menu contains options to analyse the contigs and to suggest experiments to solve problems, including: **Suggest Primers**, **Suggest Long Reads**, **Compressions and Stops**, and **Suggest Probes**. The *List* menu contains options for creating and editing lists. The *Assembly* menu contains various assembly options including: **Normal Shotgun Assembly**, **Directed Assembly**, **Screen Only**, **Assemble Independently**, **CAP3 Assembly**, **FAKII Assembly**, and **PHRAP Assembly**.

Views of the Data and the Interactions Between Displays

GAP4 has several interactive displays to enable users to view and manipulate their data at appropriate resolutions. The graphical displays include the **Contig Selector**, the **Contig Comparator**, the **Template Display**, the **Quality plot**, the **Restriction Enzyme Map**, and the **Stop Codon Map**. For editing aligned readings the **Contig Editor** and the **Join Editor** are used. From each of these the trace data for readings can be displayed and scrolled in register with the editing cursors. The displays in GAP4 communicate with one another and have linked cursors. For example, if the **Contig Editor** is being used on a contig that is also being viewed in a **Template Display**, the position of the **Contig Editor's** cursor will be shown in



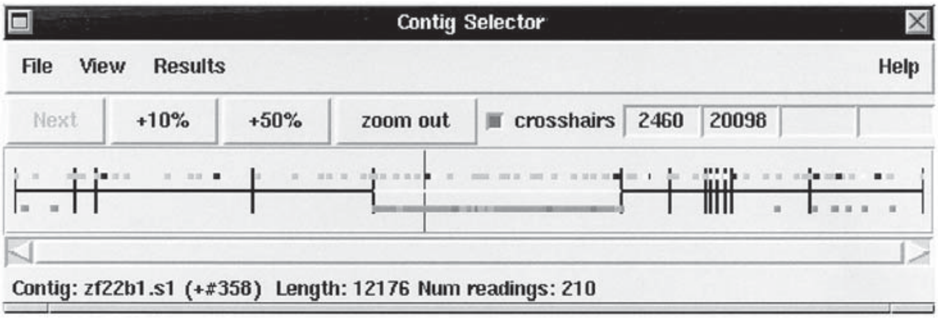
```
Selecting oligos for contig zf22b1.s1 between 1 and 12176
At  1130 - template zf38c5, primer CCAATTTCTTTCTCGTAG, number 26
At  3706 - template zf21h9, primer CTGAAAAAAGTGCTGCC, number 27
At  5521 - template zf98f5, primer GCAGGAATTGTGTGAAG, number 28
At  6252 - template zf19c6, primer GGAGGGTATCAGTGAAG, number 29
At  7751 - template zf22d11, primer TGTAATTCAAGGAAGAGG, number 30
At  8860 - template zf97c10, primer GATGATAGAAATCTCGTG, number 31
At  9799 - template zf98b5, primer GGAAAGTTGAAAGAGAAAAG, number 32
At 10333 - template zg09a9, primer CAAAACACATCATTTCGG, number 33
At 10785 - template zf24c1, primer GAGAAAGTCCTACTAATG, number 34
At 11356 - template zg29a12, primer AAAGTTTCTGGAACACGG, number 35
At 11717 - template zf97d7, primer AACCCGATTTTCGTAATG, number 36
```

Fig. 4. The main GAP4 window.

the **Template Display**, and within the **Template Display** the user can use the mouse to drag the **Contig Editor's** editing cursor. Also, if the **Contig Editor** is displaying traces, they too will scroll under the control of the **Template Display**. Any number of displays of the same contig can be shown simultaneously, including several displays of the same type, such as several **Contig Editors**. Note that the GAP4 displays make good use of color to distinguish features of the assembly.

The Contig Selector

The GAP4 *Contig Selector* (*see* Fig. 5) is used to display, select, and reorder contigs. Within the display all the contigs are shown as colinear horizontal lines separated by short vertical lines. The lengths of the horizontal lines are proportional to the lengths of the contigs and their left to right order represents the current ordering of the contigs. This contig order is stored in the GAP4 database and users can change it by dragging the lines representing the contigs in the display. The *Contig Selector* can also be used to select contigs for processing and to display tags. As the mouse cursor is moved over a contig, it is highlighted and data about it displayed in the *Information Line* at the base of the display. At the top of the display are **File**, **View**, and **Results** menus. Below these are buttons for undoing zoom-in operations and toggling the crosshair display. The four text boxes to the right show the position of the crosshair. Figure 5 shows an example of the *Contig Selector* for a project currently comprising nine large contigs and a cluster of smaller ones. The crosshair is on the largest contig and the text box to the right of the crosshair toggle button gives its position in the contig and the next box



Contig: zf22b1.s1 (+#358) Length: 12176 Num readings: 210

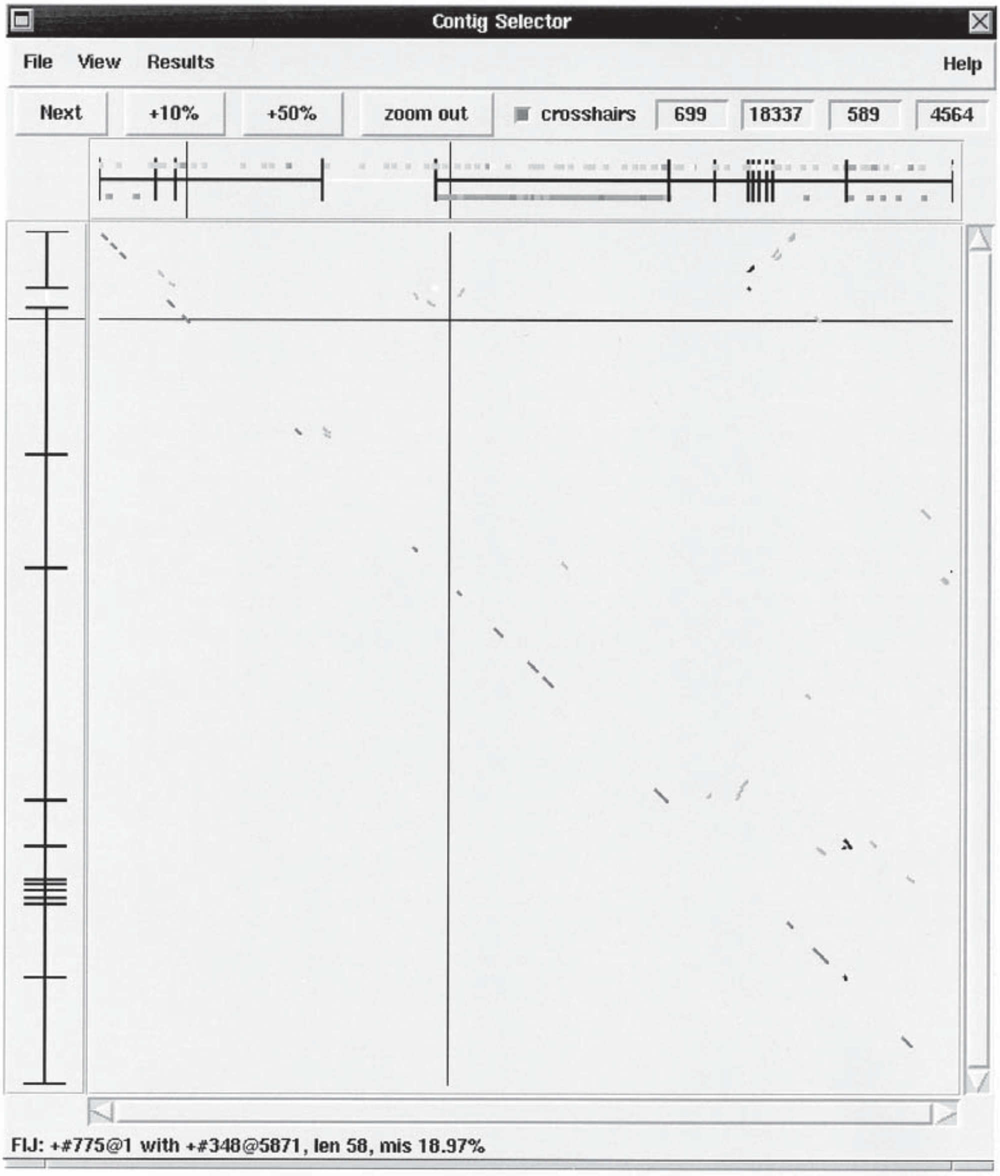
Fig. 5. The Contig Selector.

gives the position in the whole consensus. The boxes above and below the lines representing contigs show the location of tags. Those above the lines are on readings and those below are on the consensus. As can be seen the longest contig is tagged over its whole length.

The Contig Comparator

The *Contig Selector* is usually constantly in view while GAP4 is being used, but if the user selects one of the functions that in any way compares contigs, the *Contig Selector* is automatically transformed into a display known as the *Contig Comparator*. In this form it retains its *Contig Selector* functionality but can plot the results of contig comparison functions. The transformation process involves drawing an additional copy of the contig representations below and perpendicular to the original set in the *Contig Selector*, so as to produce a two dimensional area in which to plot comparative results.

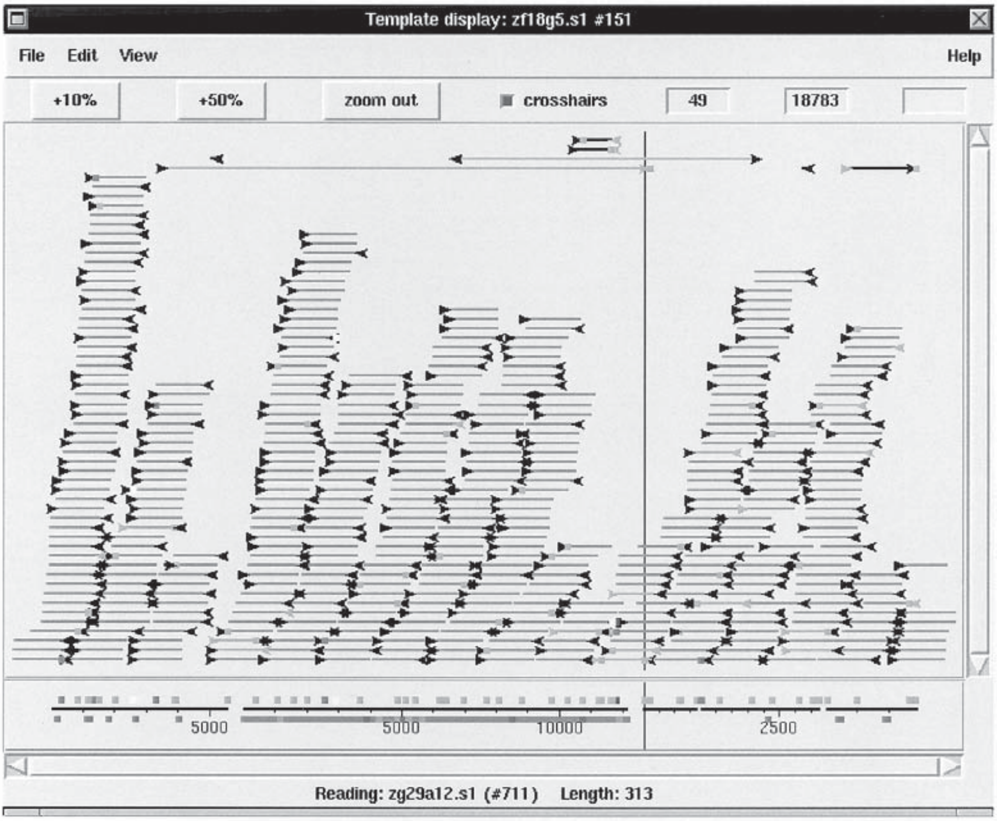
An example of the *Contig Comparator* is shown in Fig. 6. Notice that the crosshair has both vertical and horizontal components and that the position of the horizontal component is reproduced as a short vertical cursor in the *Contig Selector* part of the display. Results are not duplicated and are only plotted in the top right half of the display and on its bounding diagonal. The results plotted in this example are those from **Find Internal Joins**, **Find Read Pairs**, and **Check Assembly**; the latter are shown along the main diagonal. Obviously each type of result is drawn in a different color. Results plotted parallel to the main diagonal show relationships between pairs of contigs that are in the same relative orientation, and those perpendicular to the main diagonal are for pairs that would require one of the contigs to be complemented before they could be joined. **Find Internal Joins** compares the sequences from the ends of contigs to see if they overlap. **Find Read Pairs** locates pairs of readings that are from the same template but are in separate contigs and hence give data about the relative orientations and approximate separations of the two contigs. **Check Assembly** checks the alignment of each reading with the consensus it overlaps, and plots the positions of those of doubtful quality. This is of most value if its results are displayed at the same time as those from a search for repeats; thus, if readings with poor alignment lie on the same horizontal or vertical projections as do repeats, it may indicate that they have been assembled into the wrong copy of the repeat.



FIJ: +#775@1 with +#348@5871, len 58, mis 18.97%

Fig. 6. The Contig Comparator.

Referring to Fig. 7, note the white dot, just above and to the left of the pair of crosshairs. Its whiteness indicates that it is the result that has most recently been touched by the crosshairs. The two contigs to which the result corresponds are also shown in white, and the *Information Line* contains relevant data. It gives the identity of the two contigs, the position and length of the overlap, and the percentage mismatch of the corresponding alignment. Near to the white dot are two diagonals showing that there are two templates, each with forward and reverse readings, spanning the same pair of contigs, which according to the **Find Internal Joins** result to have an apparent



Reading: zg29a12.s1 (#711) Length: 313

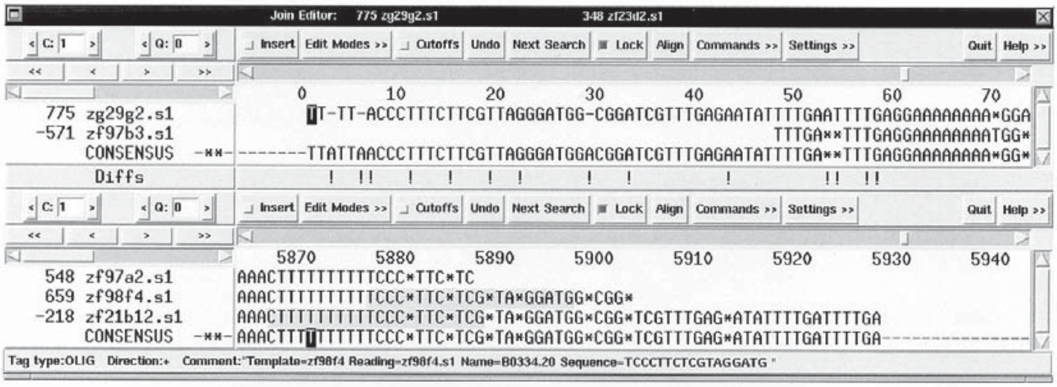
Fig. 7. Template Display of ordered contigs.

overlap. That is, there is independent evidence that these two contigs overlap and could be joined (*see* Fig. 8 and Subheading "Finding Joins Between Contigs Using Sequence Comparison").

Note that it is an interactive display. The lines representing contigs can be dragged to new locations or contigs can be complemented and their plotted results will move with them and all the others are automatically rearranged accordingly. Also, if a join is made using the **Join Editor**, the lines representing the contigs will be joined and the rest reorganized. Each type of result in the display can be operated on by a particular set of commands, and these can be invoked by moving the crosshair over the plotted line and pressing the right mouse button to pop up a menu.

The Template Display

The *Template Display* provides a graphical overview of either a set of contigs or a single contig. In our terminology, a *template* is the DNA insert from which a reading is obtained. In many laboratories, templates will be size-selected to have a length of around 2000 bases, and initial readings will be obtained from each of its ends to produce a *read-pair*. The information that can be displayed in the Template Display includes: readings, templates, tags, restriction enzyme sites, rulers, and consensus quality. As the mouse cursor is moved over any item in the display, textual data



```
                                0         10         20         30         40         50         60         70
 775 zg29g2.s1                 TT-TT-ACCCTTTCTTCGTTAGGGATGG-CGGATCGTTTGAGAATATTTTGAATTTTGAGGAAAAAAAA*GGA
-571 zf97b3.s1                                                                  TTTGA**TTTGAGGAAAAAAAAATGG*
     CONSENSUS  -**- -------TTATTAACCCTTTCTTCGTTAGGGATGGACGGATCGTTTGAGAATATTTTGA**TTTGAGGAAAAAAAA*GG*
     Diffs                    !  !!   !   !   !  !      !   !         !         !!  !!

                    5870       5880       5890       5900       5910       5920       5930       5940
 548 zf97a2.s1      AAACTTTTTTTTTTCCC*TTC*TC
 659 zf98f4.s1       AAACTTTTTTTTTTCCC*TTC*TCG*TA*GGATGG*CGG*
-218 zf21b12.s1      AAACTTTTTTTTTTCCC*TTC*TCG*TA*GGATGG*CGG*TCGTTTGAG*ATATTTTGATTTTGA
     CONSENSUS  -**- AAACTTTTTTTTTTCCC*TTC*TCG*TA*GGATGG*CGG*TCGTTTGAG*ATATTTTGATTTTGA--------------
Tag type:OLIG  Direction:+  Comment:"Template=zf98f4 Reading=zf98f4.s1 Name=B0334.20 Sequence=TCCCTTCTCGTAGGATG "
```

Fig. 8. The Join Editor.

about it will appear in the *Information Line*. The positions of a vertical crosshair, under the control of the mouse, are continuously displayed in two boxes at the top of the window. The left box shows the position in a contig and the right the position in the overall consensus. The order of the contigs can be changed by horizontally dragging the lines that represent them in the display, whereupon all their associated data will be redrawn.

In Fig. 7 we see that above the *Information Line* is a horizontal scrollbar that can be used if the plot has been magnified with the zooming mechanism. In the panel above the scrollbar are horizontal black lines representing 5 contigs, the longer ones having rulers marked on them. The boxes above and below these lines represent the positions of the respective tags on the readings and on the consensus. The longest contig contains a tag over its entire length. The vertical line cutting the contig to its right and reaching the top of the next panel, is the crosshair. This main panel contains lines representing individual templates and the readings derived from them. At the scale shown, the extent of each reading cannot be seen, and only their forward readings and gray-colored reverse readings arrow heads are visible. The templates are color coded: dark blue have only forward readings; red have both forward and reverse readings, are within the expected size range, and are in the same contig; yellow have both forward and reverse readings, in the correct relative orientation to one another, but which are in different contigs, and are within the expected size range; greenish yellow (*see* Fig. 7 on companion CD) have both forward and reverse readings, are in different contigs, but are outside their expected size range. These contigs have been processed through GAP4's automatic contig ordering function (*see* Subheading "Ordering Contigs Using Read-Pair Data"), and it may be possible to see that the gray scales of some of the templates between the contig ends are lighter, and that they have readings in each of the two contigs involved. Problem templates are separated out at the top of the display. With this ordering of the contigs, it can be seen that two of these problem templates would need to be many thousands of bases in length. In the panel containing the crosshair position boxes (the empty box is used to show the separations of restriction enzyme sites when their data are displayed), is a button for switching the crosshair on and off and a **zoom out** button, which reverts to the previous level of zoom when clicked. At the top of the display are menus that provide for closing the window, saving a changed contig order and selecting the features to be displayed.

Assembly

GAP4 can use several assembly engines including: *Normal shotgun assembly*, *Directed assembly*, *CAP3*, *FAKII*, and *PHRAP*. Particularly for repetitive data, the most reliable assembly will be achieved by global algorithms such as CAP3, FAKII, and PHRAP as they compare every reading against every other reading prior to aligning them into contigs. Later in a project, when new readings have been obtained to solve particular problems, *Directed Assembly* can be used (since their approximate location in the assembly is known). Obviously this can be very helpful if the readings have been determined to help with a difficult repeat; *Directed Assembly* will ensure that they are aligned in the correct places. Even if a global algorithm is used for assembling the initial shotgun, the *Normal shotgun assembly* algorithm is useful to enter readings obtained to help finish a project.

Ordering Contigs Using Read-Pair Data

GAP4 contains an option for automatic ordering of contigs based on read-pair data. The algorithm tries to find the most consistent orientations and left to right order for all the contigs in the database. The task is made more difficult by errors in the data owing to misnaming of readings and templates, which is usually caused by lane-tracking errors when processing the original gel data. This is becoming less of a problem with the increasing use of capiliary-based sequencing instruments. The output from the function is a list containing records giving the left to right orderings. When there are several records, this means that there is insufficient data to link all the contigs. If the user double-clicks on any of these records a *Template Display* showing their ordering will appear (*see* Fig. 7). Within this, if necessary, the order can be altered by using the mouse to move the lines representing the contigs.

Checking the Correctness of Assemblies

Prior to ensuring that the consensus sequence reaches a given standard of reliability it is advisable to check, at a coarser level, that readings have been correctly positioned. Within GAP4 there are several functions for checking on the reliability of an assembly and they all display their results in the *Contig Comparator* described earlier. We can ask: Is the read-pair data consistent? Do the quality clipped segments of readings align well with the consensus? Do the visible parts of readings align well? If not, do they match elsewhere?

As will be seen, the details of the read-pair data are most easily checked in the *Template Display*, but if these results are displayed in the *Contig Comparator* they can be viewed at the same time as other types of analysis. After the sequence is apparently completed, we need an external experimental check of its integrity, such as data from a restriction enzyme digest. GAP4 can produce a map of the restriction enzyme recognition sequences contained in the consensus sequence for comparison with experimental results.

Experiment Suggestion Functions

As a follow-up to the initial shotgun phase of a sequencing project, experiment suggesting functions are designed to help automate the task of choosing experiments to join contigs and to finish the sequence to the required level of accuracy. The package contains two programs for this purpose. Having located problems, the newest, PRE_FINISH (the details of program are outside the scope of this article), performs a sophisticated analysis of the benefits of applying a range of experiments to help to complete the project. It needs to be set up to fit the particular modes of operation of the laboratory where it is installed and hence is highly configurable. However, once set up, it is very effective.

An older and simpler set of experiment suggestion functions is available within GAP4. Here the options include **Suggest Primers**, **Suggest Long Gels**, **Suggest Probes**, and **Compressions and Stops**. One difference between these routines and PRE_FINISH is that in GAP4 the user must select the experiments type, whereas PRE_FINISH will choose the best of several types of experiments to solve the problems in each region. Figure 4 shows how the results are written to the GAP4 Output window. All of the experiment suggestion functions start by calculating the equivalent of a consensus sequence for the selected contigs, while at the same time encoding which segments contain data only on one strand. This encoding is then used to suggest suitable experiments, of the type requested by the user, to cover the single-stranded segments and extend the contigs. In the case of **Suggest Primers**, the function also adds the tag of *primer*, containing the template name and primer sequence, to the relevant reading. The **Contig Editor** also contains a primer selection function.

Finding Joins Between Contigs Using Sequence Comparisons

No assembly engine will make all the possible joins between contigs. As already mentioned, via the *Contig Comparator* GAP4 provides the user with views of the data that can reveal the safety or otherwise of making joins. The **Find Internal Joins** function compares the consensus sequences of the contigs in the database in both orientations to detect possible overlaps. It works in two modes: **quick** and **sensitive**. The quick mode is very rapid and should be applied before the sensitive mode is used to find any remaining overlaps. During large-scale sequencing where overlapping BAC clones are being sequenced, it is common practice to exchange readings from their GAP4 databases to help finish the projects. In this case **Find Internal Joins** may be employed to join the overlap between the two sets of readings. Here the quick mode is particularly useful, as it can find and align the overlaps between 100,000 base consensus sequences in a few seconds. The sensitive mode is slower, but will find weak matches between the ends of contigs.

For both modes, a number of options are available. The user can select to extend the contig consensus using segments of the hidden data near the contig ends. In conjunction with the sensitive mode this should find all possible joins. Users can also choose to mask any tagged regions in the contigs: typically regions tagged as being ALU or other repeat sequences. The aligments are displayed in the *Output Window* and the results are plotted in the **Contig Comparator** (*see* Fig. 6). A **Next** button will appear at the top of the **Contig Comparator**. The results are sorted in descending order of quality, and each time the user clicks on the **Next** button, the next best possible join will appear in a **Join Editor** window (*see* Fig. 8). As an alternative to the **Next** button, users can click on the results plotted in the **Contig Comparator** to bring up the **Join Editor**. The **Join Editor** is two **Contig Editors** sandwiched together with a horizontal strip between (labeled *Diffs*) to indicate disagreements in the consensus sequences. The **Join Editor** also has **Align** buttons for aligning the two consensus sequences and a **Lock** button for forcing the two contigs to scroll together. The join shown in Fig. 8 is obviously very poor, but it is confirmed by read-pair data as was discussed in the section on the *Contig Comparator*. Within the **Join Editor** users should align the contigs, making any necessary edits, and checking the trace data. When they exit the editor they can choose to make or reject the join. The **Join Editor** contains all the functionality of the **Contig Editor**, which is described next.

Checking and Editing Contigs with the Contig Editor

The GAP4 contig editor is designed to allow rapid checking and editing of characters in assembled readings. Very large savings in time can be achieved by its sophisticated problem-finding procedures, which automatically direct the user only to the bases that require attention.

Contig Editor Search Modes

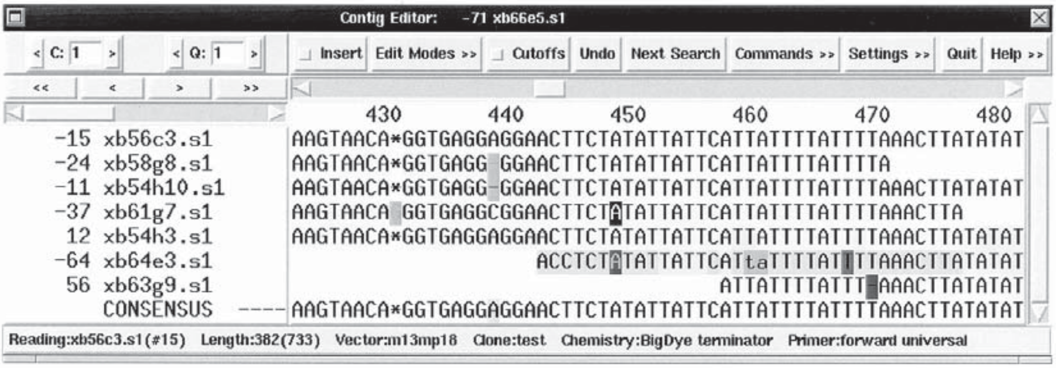
The search modes available include the following:

1. **Search by problem** moves the cursor to the next position in the consensus that is not A,C,G, or T. The character in the consensus will depend on the consensus algorithm being employed and the thresholds set.
2. **Search by quality** moves the cursor to the next position in the sequence where the consensus characters, calculated separately for the two strands, disagree.
3. **Search by consensus quality** moves the cursor to the next position in the consensus where the confidence is below a set threshold.
4. **Search by discrepancies** moves the cursor to the next position where two readings contain aligned but disagreeing bases above a given confidence value.
5. **Search by edit** moves the cursor to the next position where an edit has been made, hence allowing all previous edits to be checked.
6. **Search by evidence 1** moves the cursor to the next edited position for which the consensus base is not supported by any evidence from the original data.
7. **Search by evidence 2** moves the cursor to the next edited position for which the consensus base is only supported by evidence from the original data on one of the two strands. *Search evidence* means that a base of the same type must appear at an equivalent position in an original, unedited reading.

These searches allow rapid checking of all the edits made. The other search modes provide more obvious operations such as: moving to a particular position, to a particular read, to the next occurrence of a given sequence, to the next tag of a given type, and to the next tag containing a given text string. It is also possible for the search to be controlled by a file containing commands.

The Contig Editor Window

In the *Contig Editor* window (*see* Fig. 9), reading numbers and names are on the left, sequences to their right, status lines are at the bottom, and controls are at the top. The buttons labeled **Settings**, **Commands**, and **Edit modes** reveal tear-off menus, and their contents are described below. The **Commands** menu contains a **Search** dialogue from which a search method appropriate to the data and current task can be selected. The editor display includes color coding of the base confidence values in the readings and in the consensus using gray scales, and marking of all the edits made; padding characters in green, changes in pink, deletions in red, and confidence value



```
                           430       440       450       460       470       480
   -15 xb56c3.s1          AAGTAACA*GGTGAGGAGGAACTTCTATATTATTCATTATTTTATTTTAAACTTATATAT
   -24 xb58g8.s1          AAGTAACA*GGTGAGG GGAACTTCTATATTATTCATTATTTTATTTTA
   -11 xb54h10.s1         AAGTAACA*GGTGAGG-GGAACTTCTATATTATTCATTATTTTATTTTAAACTTATATAT
   -37 xb61g7.s1          AAGTAACA GGTGAGGCGGAACTTCTATATTATTCATTATTTTATTTTAAACTTA
    12 xb54h3.s1          AAGTAACA*GGTGAGGAGGAACTTCTATATTATTCATTATTTTATTTTAAACTTATATAT
   -64 xb64e3.s1                              ACCTCTATATTATTCATtaTTTTATTTTAAACTTATATAT
    56 xb63g9.s1                                             ATTATTTTATTT-AAACTTATATAT
       CONSENSUS     ---- AAGTAACA*GGTGAGGAGGAACTTCTATATTATTCATTATTTTATTTTAAACTTATATAT
Reading:xb56c3.s1(#15)  Length:382(733)  Vector:m13mp18  Clone:test  Chemistry:BigDye terminator  Primer:forward universal
```

Fig. 9. The Contig Editor.

changes are highlighted in blue. Each time the **Next search** button is clicked, a search of the selected type is performed to locate the next problem in the sequence requiring visual inspection by the user. The **Undo** button will successively reverse all edits up to the last save. The **Cutoffs** button toggles on and off the visibility of the *hidden* data at the ends of readings. The final button along the top of the editor, **Insert**, toggles between insert and replace modes. The two text boxes with < and > symbols next to them are for altering the thresholds for the consensus algorithms. Just below these are more buttons that can be used to scroll the edit window. Below these is a scrollbar for viewing long reading names. The *Information Line* reveals data about the lengths of readings and the names of their vectors, or tags.

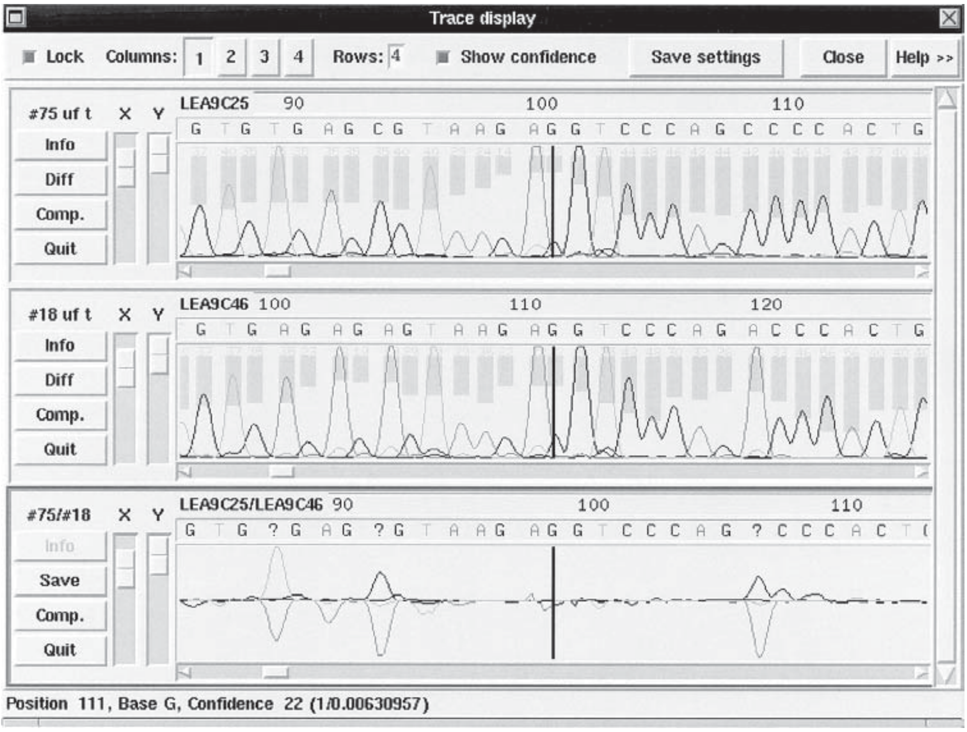
Editing Modes and Settings

Different laboratories have different working practices, and some wish to allow only particular people to perform certain types of editing operations. The **Contig Editor Edit modes** menu contains the following editing operations: 1) insert in read, delete in read, insert in consensus, delete dash (undetermined) in consensus, 2) delete any character type in consensus, 3) replace in consensus, reading shift, and 4) transposition.

Each of these may be switched on or off individually. The editor also has two preconfigured sets of edits: **Mode 1** and **Mode 2**, which define for example, a limited, safe set, and a powerful set for a more senior operator. Note that during editing the consensus is updated immediately for each edit made.

The **Contig Editor Settings** menu contains the following: **Status Line**, **Trace Display**, **Highlight Disagreements**, **Compare Strands**, **Toggle auto-save**, **3 character amino acids**, **Group Readings by Templates**, **Show reading quality**, **Show consensus quality**, **Show edits**, **Set Active Tags**, **Set Output List**, and **Set Default Confidences**. For example, the Status line can be configured to show if the contig has data on both strands, whether edits have been made, whether to show translation to amino acids. The **Trace Display** can be configured to automatically show traces relevant for each problem as revealed by the search command, by defining the number of rows and columns of traces to show at once.

The **Contig Editor Commands** menu includes commands: to operate on tags, to save the edits, to save the contents of the **Contig Editor** display as a text file, to create a *consensus trace*, and save it to a file; to select primers to align segments of readings,


Trace display
Lock
Columns: 1 2 3 4
Rows: 4
Show confidence
Save settings
Close
Help >>
#75 uf t
X Y
Info
Diff
Comp.
Quit
LEA9C25 90 100 110
G T G T G A G C G T A A G A G G T C C C A G C C C C A C T G
#18 uf t
X Y
Info
Diff
Comp.
Quit
LEA9C46 100 110 120
G T G A G A G A G T A A G A G G T C C C A G A C C C A C T G
#75/#18
X Y
Info
Save
Comp.
Quit
LEA9C25/LEA9C46 90 100 110
G T G ? G A G ? G T A A G A G G T C C C A G ? C C C A C T (

Position 111, Base G, Confidence 22 (1/0.00630957)

Fig. 10. The Trace Display including a difference plot.

to line up and remove padding characters, to remove readings, to break the contig, and to search the contig.

The Trace Display

Within the **Contig Editor** the **Trace Display** read can be invoked by double-clicking on a *base* in the read. By double clicking on the consensus, the traces for all the readings at that point will be displayed. An example of the GAP4 **Trace Display** is shown in Fig. 10. Here it has been configured to show up to four rows in one column, but it can be convenient to use more rows and multiple columns. In this example it is being used to show a pair of traces, with a plot of their differences below. For each reading the display can show the fluorescence amplitudes, the base calls and, as an inverted histogram, their confidence values. When the **Contig Editor** cursor is moved the traces scroll in register. Conversely, the **Contig Editor** can be scrolled from within the **Trace Display**. As shown here, individual confidence values can be shown on the Information Line at the base of the display and textual data about each reading can be viewed in a separate window. The traces can be zoomed in x and y using the scrollbars at the left edge. The difference plot is invoked using the **Diff button**. As can be seen in this example (bottom panel), in most positions the *signal* is quite low, but in three places there are coincident peaks and troughs, indicating possible mutations. Here these mutations are borne out by the base calls, but as automated in our trace_diff program, such analysis has been very helpful in distinguishing mutations from sequence differences due to base calling errors.

Creating a Consensus Sequence

GAP4 can calculate a consensus sequence and save it to a disk file in *FASTA* or *Experiment* file formats. *FASTA* is useful for programs such as *BLAST*. *Experiment* file format is very close to that of EMBL sequence library entries and can contain the text fields of selected tags added by the user during processing. If users follow our recommendation of making as few changes to their data as possible, the consensus will contain many padding characters, but users can choose to have these removed before the consensus sequence is written to the disk.

Glossary and Abbreviations

BAC Bacterial artificial chromosome used to clone segments of DNA up to 200 kbp.

Base Caller A computer program which interprets the fluorescence intensitites from a sequencing instrument to determine its sequence.

Basecall Confidence Values Numerical values assigned to each base in a sequence reading to predict its reliability. Confidence C = –10*log10(probability of error)

Consensus Sequence Multiple sequence alignments can be represented by a single consensus sequence derived by analyzing the frequency and confidence of the different character types aligned at each position.

Contig An ordered set of overlapping segments of DNA from which a contiguous sequence can be derived.

DNA Sequencing Traces In DNA sequencing, the chains of nucleotides are labeled with fluorescent dyes. The intensity of fluorescence measured as DNA fragments are electrophoresed in a sequencing instrument known as a chromatogram, or, more commonly, trace. The traces are processed to produce the DNA sequence.

DNA Template The segments of DNA inserted into a vector for sequencing. Also known as “fragments” and “inserts.”

Experiment Files A file format used to transfer data between sequence assembly processing programs.

Forward Read A DNA sequence obtained from a single sequencing experiment using a primer for the forward strand of DNA.

Gap 4 A computer program for handling DNA sequencing projects.

PREGAP 4 A computer program for preparing DNA trace data for sequence assembly using gap4.

Primer A short (approx 20 bases) segment of single-stranded DNA of known sequence. The Sanger sequencing method and the Polymerase Chain Reaction are both initiated by a primer annealed to the DNA template.

Read-Pair A pair of sequence readings obtained from opposite ends of a DNA sequencing template.

Reading (Gel Reading) A DNA sequence obtained from a single sequencing experiment, typically around 500-800 bp.

Reverse Read A DNA sequence obtained from a single sequencing experiment using a primer for the reverse strand of DNA.

SCF A widely used, non-proprietary file format for storing DNA trace data.

Sequence Assembly The process of arranging a set of overlapping sequence readings into their correct order along the genome.

Sequencing Vectors Vectors designed to aid DNA sequencing.

VECTOR A DNA molecule into which foreign DNA can be inserted for introduction into a new host where it will be replicated, thus producing mass quantities. Depending on the type of vector, foreign DNA of lengths between 1 kb and 2 mb can be inserted.

ZTR The newest and most compact file format for storing trace data.

Suggested Readings

Sequence Assembly

Bonfield, J. K., Smith, K. F., and Staden, R. (1995) A new DNA sequence assembly program, Nucleic Acids Res. 23, 4992–4999.

Green, P. H. (1997) Personal communication.

Huang, X. and Madden, A. (1999) CAP3: A DNA Sequence Assembly Program, Genome Res. 9, 868–877.

Myers, E. W., Jain, M., and Larson, S. (1996) Internal report, University of Arizona, Tucson, AZ.

Staden, R. (1980) A new computer method for the storage and manipulation of DNA gel reading data, Nucleic Acids Res. 8, 3673–3694.

File Formats: SCF, ZTR, Experiment, Fasta Files, and the GAP4 Database

Bonfield, J. K. and Staden, R. (2002) ZTR: A new format for DNA sequence trace data, Bioinformatics 18(1), 3–10.

Bonfield, J. K. and Staden, R. (1996) Experiment files and their application during large-scale sequencing projects, DNA Sequence 6, 109–117.

Dear, S. and Staden, R. (1992) A standard file format for data from DNA sequencing instruments, DNA Sequence 3, 107–110.

Pearson, W. R. (1994) Using the FASTA Program to Search Protein and DNA Sequence Databases, in: Methods in Molecular Biology, (Griffin, A. M. and Griffin, H. G., eds.), vol. 25, Humana Press, Totowa, NJ, pp. 365–389.

Tool Kits

Free Software Foundation, 675 Mass Avenue, Cambridge, MA 02139.

Ousterhout, J. K. (1994) Tcl and the Tk toolkit, Addison-Wesley Publishing Company, Reading, MA.

Other Programs

Haas, S., Vingron, M., Poustka, A., and Wiemann, S. (1998) Primer design for large scale sequencing, Nucleic Acids Res. 26, 3006–3012.

Bonfield, J. K., Rada, C., and Staden, R. (1998) Automated detection of point mutations using flourescent sequence trace subtraction, Nucleic Acids Res. 26, 3404–3409.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool, J. Mol. Biol. 215, 403–410.

21 OLIGO Primer Analysis Software

John D. Offerman and Wojciech Rychlik

Introduction

Concurrent with the development of the polymerase chain reaction (PCR) in the late 1980s as an essential technique in the molecular biology laboratory, optimal design of PCR primers became an important part of the process. It was soon learned, for example, that inattention to secondary structure in synthetic primers could create primer-dimers and hairpins and could seriously reduce or eliminate the yield of the PCR product. The presence of priming sites in the template other than the intended target—so called *false priming* sites—were also found to interfere with PCR efficiency, by generating unwanted PCR products and *background* on the gel. Furthermore, the efficient PCR experiment required the proper concentrations of salt, buffer, and nucleic acid and the accurate determination of melting temperatures of the primers and the template. Cumbersome calculations were required to produce these values and their complexity often resulted in errors.

With limited success, PCR experimentalists trained their eyes to spot primer dimers, hairpins, and false priming sites in their DNA sequence. Many came to believe that they were exceptionally good at spotting these problems, when, in fact, it was the nature of PCR to proceed, after a fashion, despite poor oligonucleotide design. But most researchers understood early on that the development of a computer program was needed to quickly determine accurate values of oligonucleotide melting temperature (Tm) and to check the oligonucleotide sequence for the presence of secondary structure and PCR targets for the presence of false priming sites.

The first widely used program, OLIGO version 2, was developed by Dr. Wojciech Rychlik while doing PCR research at the University of Kentucky, and offered free through Nucleic Acids Research in 1989. The first commercial program, OLIGO v. 3 for the PC-DOS operating system, was released by Dr. Rychlik in 1989 and distributed through National Biosciences, Inc. (Minneapolis, MN). Version 3 provided both secondary structure and false priming analysis functions, and also aided the researcher in the design and selection of oligonucleotides for DNA sequencing.

Since version 3, there have been a number of competitive commercial applications offered, including several PCR/primer selection modules integrated into larger DNA sequence-analysis programs. These have been theoretical and algorithm-driven.

OLIGO's selection algorithms and assumptions have always been grounded in PCR laboratory research by the author and others, and fully referenced in the OLIGO manual.

Since OLIGO version 3, there have been three more PC as well as three more Macintosh releases of the program. The current OLIGO products are: version 6 for Windows, first released in 1999, and version 6 for Macintosh, released in 1998. New major versions are scheduled for release every 3 years. Single-user licenses provide for use on one to four computers under one principal investigator. Program network licenses are also available that offer from one to any number of simultaneous users of the program. There have been many new processes and procedures introduced to the molecular biology laboratory in the last 15 years, including many improvements to, and permutations of, PCR and DNA sequencing. OLIGO has, concurrently, added a range of new features that aid the researcher in improving the performance of these new tools. The features of OLIGO version 6, why they were needed, and how they simplify and optimize the selection and use of oligonucleotides in the laboratory, will be discussed in this chapter.

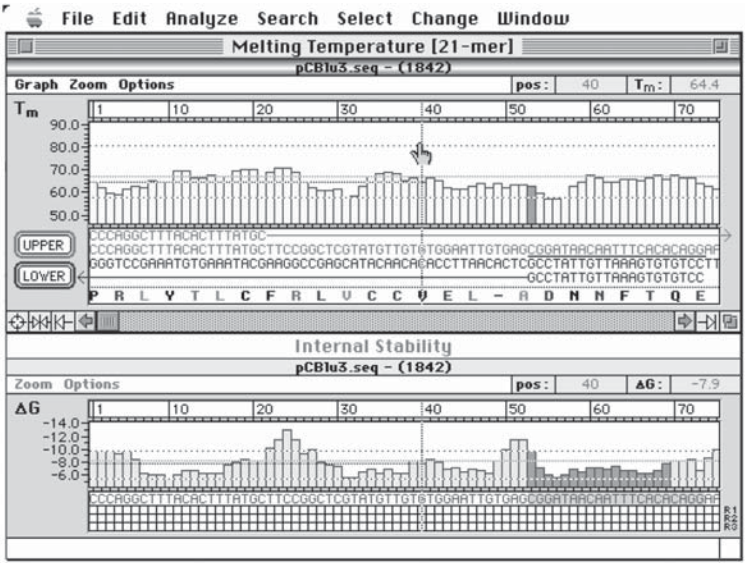
Program Organization

OLIGO 6 is organized into several major features and functions: the analysis functions under the **Analyze** menu, the search functions under the **Search** menu, edit/translation functions under the **Edit** menu, the oligo database functions, and the order form. The *Melting Temperature* and the *Internal Stability* windows (*see* Fig. 1), are the main analysis windows that open when a sequence file is loaded. These windows are the organizational center of the program. The melting temperature (Tm) window displays both DNA strands of the sequence, the translated protein sequence, and a histogram of the melting temperatures of every theoretical oligonucleotide in the file. Any oligo can be quickly analyzed in this window as well by selecting it as the **current oligo**, clicking the mouse on the **5' terminus**, and then applying the analysis functions. The current oligo can be selected as the upper (forward) primer by clicking the red **UPPER** button or lower (reverse) primer by clicking the **LOWER** button. Also, when primers are selected following the automated primer/probe searches, they are displayed as upper (or lower) primers in the melting temperature window.

The *Internal Stability* window displays free energy of DNA or RNA pentamers throughout the sequence visualizing GC clamps and AT-rich low Tm areas. It also displays the positive strand sequence and marked search-selected oligonucleotides, if any, on the bottom. The *current oligo* sequence, available in the **Analysis** menu, is highlighted (*see* Fig. 1).

PCR Primer Search and Selection

PCR has proven to be an amazingly robust laboratory procedure over the years, and has found a prominent place in most laboratories working with nucleic acids. However, there are a number of oligonucleotide design and other factors that can interfere with the optimization of the PCR process. OLIGO 6 searches DNA and RNA sequences to eliminate from consideration oligonucleotides that exhibit one or more of these interfering factors. Specifically, the program performs as many as nine different subsearches, eliminating oligonucleotides in each subsearch that fail to meet the threshold set by the program or the user. Potential PCR problems due to oligonucleotide design, and the subsearches that OLIGO uses to discard oligonucleotides containing these flaws, are discussed below.



```
CCCAGGCTTTACACTTTATGC
CCCAGGCTTTACACTTTATGCTTCCGGCTCGTATGTTGTGTGGAATTGTGAGCGGATAACAATTTCACACAGGAA
GGGTCCGAAATGTGAAATACGAAGGCCGAGCATACAACACACCTTAACACTCGCCTATTGTTAAAGTGTGTCCTT
                                                    GCCTATTGTTAAAGTGTGTCC
P  R  L  Y  T  L  C  F  R  L  V  C  C  V  E  L  –  A  D  N  N  F  T  Q  E

CCCAGGCTTTACACTTTATGCTTCCGGCTCGTATGTTGTGTGGAATTGTGAGCGGATAACAATTTCACACAGGAA
```

Fig. 1. The Melting Temperature and Internal Stability windows.

The most serious oligonucleotides design deficiency for PCR is certainly the stable 3' dimer, where the 3' end of the PCR primer binds strongly to itself or to its companion primer. If a 3' dimer is sufficiently stable so that it preferentially hybridizes to itself or its mate, rather than to the intended target on the template, the PCR product yield can be reduced essentially to zero.

Hairpin structures in oligonucleotides can also greatly reduce PCR yield. OLIGO discards primers with 3' dimers or hairpins (duplexes) based on the stability of the secondary structure, measured by the G value, or alternatively, by the length of contiguous matching nucleotides. Because PCR has been shown to be adversely affected by 3' duplexes, OLIGO also includes a setting for the number of nucleotides on the 3' end to be checked (so as not to eliminate otherwise suitable oligonucleotides containing harmless 5' dimers). OLIGO performs the above checks in the **Duplex-free Oligonucleotides** subsearch. Another subsearch specifically for hairpins is also executed; this subsearch considers loop size while testing the entire set of oligonucleotides for hairpins. This subsearch is important in the search for *Hybridization Probes*.

OLIGO then searches for and discards *highly specific* oligonucleotides. Research by the author has determined that oligonucleotides that are too stable on their 3' ends (irrespective of dimer formation), are more likely to false prime randomly on unknown nucleic acid templates. This is because a shorter number of matching nucleotides on a stable 3' end are required to initiate priming on a false target than are required with a less stable 3' end. Statistically, a stable 3' end will create more false priming and unwanted background on the PCR gel.

The early convention was to design oligonucleotides with a *GC clamp* on the 3' end, in order to *clamp down* the oligo on its target. This practice would add stability but compromise specificity. OLIGO selects primers that are both specific and stable. It does this by employing subsearches that: 1) discard oligos that have excessively stable 3' ends, and 2) discard oligonucleotides that do not have one or more stable sections (a GC clamp) at the 5' end or central segment of the oligonucleotide. So the remaining oligonucleotides, following the **Highly Specific Oligos** subsearch and the **GC Clamp** subsearch, are both specific to their target and will hybridize with sufficient stability to permit priming.

False priming can create unwanted background, but are particularly problematic when they occur in or near the intended PCR product. OLIGO's false priming subsearch compares the 3' end of each potential primer with the active sequence file and discards any that are found to have false priming sites. Using another false priming subsearch, the researcher can also check each potential primer against other DNA or RNA sequence files on their computer. This is the **Continue False Priming Search in Other Files** subsearch. OLIGO employs a novel algorithm, developed experimentally, called the *priming efficiency* calculation, to predict priming and false priming. Priming efficiency (PE), determines whether an oligonucleotides will prime on a given site under a given set of conditions. This important feature of the OLIGO program is discussed on p. 351 (*see* Subheading "Priming Efficiency Measurements for Priming and False Priming").

The next subsearch used in OLIGO's PCR primer search is for *Oligonucleotides within Selected Stability Limits*. Oligonucleotides that have a high, but not excessively high, Tm, work best in PCR reactions. This subsearch discards all oligonucleotides that fall outside this Tm window. Another important feature of optimal PCR primers is the matching of Tms. False priming and unnecessarily high annealing temperatures can result when a PCR pair is selected having an excessive Tm difference. OLIGO automatically matches Tms of primer pairs following the PCR search and selection process by dropping nucleotides from the primer with the higher Tm until the Tms or priming efficiencies are in close agreement.

Homooligomers (i.e., GGGGG) and sequence repeats (i.e., GCGCGC), are eliminated in the **Eliminate Oligomers/Sequence Repeats** subsearch. Homooligomers and sequence repeat structure in primers can cause misalignment on the target and subsequently impair annealing kinetics or even cause synthesis of PCR products of different size.

The final subsearch of the PCR primer selection process, **Eliminate Frequent Oligos**, discards oligonucleotides that have 3' ends that occur frequently in the GenBank nucleic acid database. OLIGO 6 includes a table of frequencies of all 4,096 or 16,384 combinations of 6 or 7 nucleotide segments, respectively. When this subsearch is run, the 6-mer (or 7-mer) 3' ends of oligonucleotides in the active sequence are checked against the frequency table. As with the **Highly Specific Oligos** subsearch, this search further reduces the likelihood of false priming and background when using genomic DNA for PCR analysis.

However, when the PCR template is not to be run in genomic DNA or other complex substrates, the user can click off the **Complex Substrate** box in the search parameters window. This turns off the **Highly Specific Oligos** and **Frequent Oligos** subsearches, thereby saving otherwise suitable primers for use in plasmid DNA and other simple substrates.

When the researcher loads his/her nucleic acid sequence file of interest and selects the **Search for Primers and Probes - PCR Primers: Compatible Pairs**, the program activates the above subsearches and once the search is initiated, they are sequentially executed. When the full (composite) PCR search is complete, only those oligos that have met or exceeded the threshold settings for each subsearch are selected.

The upper (forward) and lower (reverse) primers are then compared for cross-compatibility (no 3' end interaction) and all cross-compatible pairs are compiled in the *Primer Pairs* table. Clicking on a given pair in the table brings up the PCR window, which displays PCR conditions, Tms, and other data for those primers, including a proportional schematic of the PCR product on the template. Primer pairs in the table can be sorted by position on the template, PCR product size, GC content, or PCR optimal annealing temperature. The functioning of each subsearch used in the PCR primer search depends, of course, on the threshold settings of the search parameters by which they are controlled. Because of the complexity of these parameters and the challenge of knowing where they should be set for any given search, the program includes an **expert system**. The program also allows the user to simply select one of 6 search stringencies ranging from *very high* to *very low*. The program will then insert the settings for each search parameter, based on the current understanding of the PCR process. However, users can set their own values for one or more of the search parameters and lock them in, overriding the expert system. In this way the program has maximum flexibility and adaptability to the users particular needs, while still being very easy to use.

Depending on the size and nature of nucleic acid sequences, it is hard to predict how many optimal primer pairs can be returned from a given search. The program includes a feature that will automatically drop the search stringency down one level and search again, if no primer pairs are found in the first search. A complete record of each PCR primer search is saved in the *Search Results* window. This includes all search settings and the specific results of each subsearch and primer pairings.

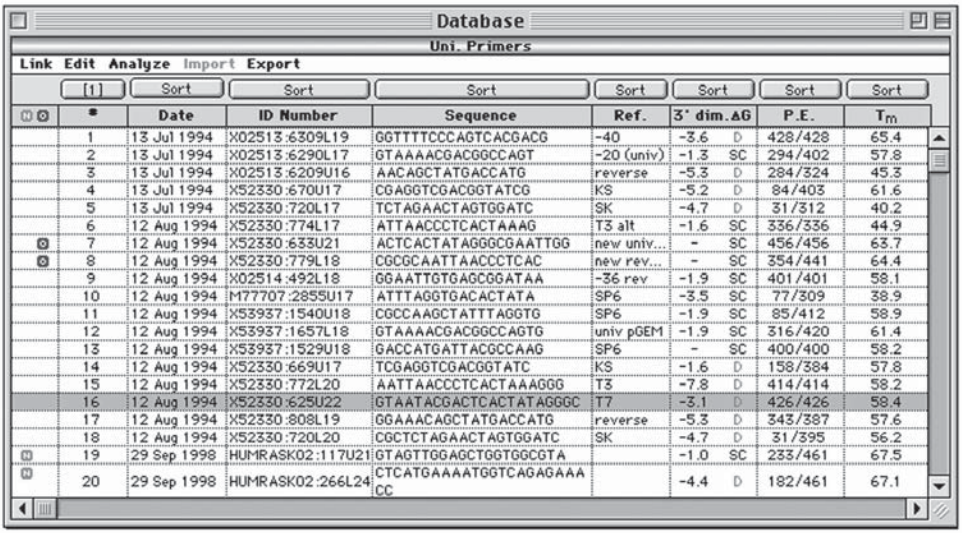
The Oligonucleotide Database and Order Form

The modern PCR laboratory can design and synthesize hundreds of primers per week or more. Keeping track of them all and their target templates can be a challenge, and transferring oligo sequences to synthesis providers can be cumbersome and time-consuming. The oligonucleotide database and order form features in OLIGO 6 can save researchers time and minimize transposition and data-maintenance errors. When a primers and probes search is selected in the OLIGO program, the primer database can be downloaded in a single operation into an oligonucleotide database window, using the **Import** function. Further, an entire group of primers selected in an OLIGO search can be imported in a single operation by using the **Multiplex Primers** option, as shown in Fig. 2.

When oligonucleotides are imported into a database, they are automatically dated and named. The naming convention is useful in keeping track of important oligonucleotide characteristics. The name includes the sequence file (template) name or the GenBank accession number, the position of the oligonucleotides on the template, whether it's an upper or lower primer, and its length.

In the database, the program also reports an oligonucleotide's Tm, its 3' end dimer status, its sequence, priming efficiency, and whether they are new entries in the database. There is also a reference field for the researcher to record pertinent notes. This window also allows for primer multiplexing, as discussed in the following.

Oligonucleotides in the database can be selected for export to an oligonucleotide order form. In the order form, the researcher can set up a template that includes ship to:/bill to: ordering information, including such specifications as synthesis scale, puri-



N ⊙	*	Date	ID Number	Sequence	Ref.	3' dim.ΔG	P.E.	T_m
	1	13 Jul 1994	X02513:6309L19	GGTTTTCCCAGTCACGACG	-40	-3.6 D	428/428	65.4
	2	13 Jul 1994	X02513:6290L17	GTAAAACGACGGCCAGT	-20 (univ)	-1.3 SC	294/402	57.8
	3	13 Jul 1994	X02513:6209U16	AACAGCTATGACCATG	reverse	-5.3 D	284/324	45.3
	4	13 Jul 1994	X52330:670U17	CGAGGTCGACGGTATCG	KS	-5.2 D	84/403	61.6
	5	13 Jul 1994	X52330:720L17	TCTAGAACTAGTGGATC	SK	-4.7 D	31/312	40.2
	6	12 Aug 1994	X52330:774L17	ATTAACCCTCACTAAAG	T3 alt	-1.6 SC	336/336	44.9
⊙	7	12 Aug 1994	X52330:633U21	ACTCACTATAGGGCGAATTGG	new univ...	- SC	456/456	63.7
⊙	8	12 Aug 1994	X52330:779L18	CGCGCAATTAACCCTCAC	new rev...	- SC	354/441	64.4
	9	12 Aug 1994	X02514:492L18	GGAATTGTGAGCGGATAA	-36 rev	-1.9 SC	401/401	58.1
	10	12 Aug 1994	M77707:2855U17	ATTTAGGTGACACTATA	SP6	-3.5 SC	77/309	38.9
	11	12 Aug 1994	X53937:1540U18	CGCCAAGCTATTTAGGTG	SP6	-1.9 SC	85/412	58.9
	12	12 Aug 1994	X53937:1657L18	GTAAAACGACGGCCAGTG	univ pGEM	-1.9 SC	316/420	61.4
	13	12 Aug 1994	X53937:1529U18	GACCATGATTACGCCAAG	SP6	- SC	400/400	58.2
	14	12 Aug 1994	X52330:669U17	TCGAGGTCGACGGTATC	KS	-1.6 D	158/384	57.8
	15	12 Aug 1994	X52330:772L20	AATTAACCCTCACTAAAGGG	T3	-7.8 D	414/414	58.2
	16	12 Aug 1994	X52330:625U22	GTAATACGACTCACTATAGGGC	T7	-3.1 D	426/426	58.4
	17	12 Aug 1994	X52330:808L19	GGAAACAGCTATGACCATG	reverse	-5.3 D	343/387	57.6
	18	12 Aug 1994	X52330:720L20	CGCTCTAGAACTAGTGGATC	SK	-4.7 D	31/395	56.2
N	19	29 Sep 1998	HUMRASK02:117U21	GTAGTTGGAGCTGGTGGCGTA		-1.0 SC	233/461	67.5
N	20	29 Sep 1998	HUMRASK02:266L24	CTCATGAAAATGGTCAGAGAAACC		-4.4 D	182/461	67.1

Fig. 2. Oligonucleotide Database window.

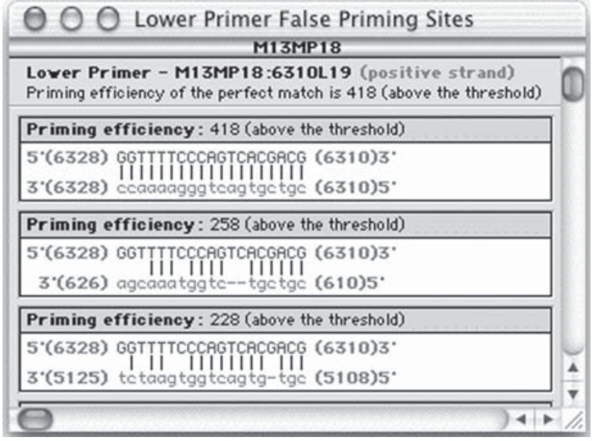
fication method, and end modifications. Then, with each synthesis order, one need only export the oligonucleotide sequences to be ordered from the database to the order form, which can then be faxed or e-mailed to the synthesis provider.

Multiplex PCR Design

PCR researchers often wish to amplify multiple PCR products in the same tube, in genetic test development or in other applications where operational efficiency is needed. In multiplex primer applications, it is necessary that all primers going into a PCR reaction be cross compatible, (dimers will not form between any 3' ends).

OLIGO 6 includes two multiplex primer selection features. One is designed for the selection of multiplex primers from a single gene or sequence file, such as the TaqMan probes selection. This multiplex selection feature is included under the **Analyze** menu. The second, more commonly used, multiplex feature provides for the selection of multiplex primers from two or more genes/exons (in separate sequence files). Users load and search target sequences for optimal PCR primers and then download them into an oligonucleotide database as described earlier. Downloading several primers can be accomplished using the **Import – Multiplex Primers** menu selection.

When primer groups from each sequence have been loaded into a database, the actual multiplex process can be initiated. The user selects a candidate primer from gene sequence #1, and then clicks **Analyze Multiplex**. This checks the 3' end of this primer against all the other primers in the database. The primers that are compatible with the multiplexed primer are identified with a *C* (in the **3' dimer G** column of the database) and noncompatible primers with *NC*, along with the numeric value of free energy (G). The second multiplex primer is then selected from the *C* (compatible) group, and multiplexed. Now, all remaining primers displaying a *C* are compatible with both primers #1 and #2. This process is continued until all the desired multiplex primers (all cross-compatible in all combinations) are selected.



```
Lower Primer - M13MP18:6310L19 (positive strand)
Priming efficiency of the perfect match is 418 (above the threshold)

Priming efficiency: 418 (above the threshold)
5'(6328) GGTTTTCCCAGTCACGACG (6310)3'
         |||||||||||||||||||
3'(6328) ccaaaagggtcagtgctgc (6310)5'

Priming efficiency: 258 (above the threshold)
5'(6328) GGTTTTCCCAGTCACGACG (6310)3'
            ||| ||||  ||||||
 3'(626) agcaaatggtc--tgctgc (610)5'

Priming efficiency: 228 (above the threshold)
5'(6328) GGTTTTCCCAGTCACGACG (6310)3'
          | ||  |||||||| |||
3'(5125) tctaagtggtcagtg-tgc (5108)5'
```

Fig. 3. False Priming Sites window. The site with priming efficiency of 258 points is the experimentally confirmed strongest false priming site of the universal M13 forward primer.

Numerous working multiplex PCR primer sets have been designed using the OLIGO 6 program. The best results in the design of large multiplexes are achieved by downloading as many optimal primers from each gene sequence as possible. Quite frequently, certain multiplex primer selections will prove to be incompatible with too many of the other primers in the database. When this occurs, the experienced user will deselect that problem primer and select another.

Priming Efficiency Measurements for Priming and False Priming

In early versions of OLIGO, and in most other programs even today, the search for false priming sites was (and is) carried out by comparing the 3' ends of potential primers with the template sequence to identify exact complementarity of 4, 5, 6, or (n) nucleotides. While this remains a simple search to perform, it is of limited use in determining whether a primer will actually false prime at a given site. For example, three complementary bases at the 3' end, followed by a mismatch, followed by 4 more complementary bases would be missed by the program as a false priming site when, in fact, it could false prime effectively.

The concept of *priming efficiency* was introduced to provide an accurate way to predict false priming. The proprietary algorithm takes into account mismatches, bulges, and loops and their positions relative to the 3' end of the primer. Priming efficiency (PE) is simple in concept, and is represented by a single number for each primer on a given target. A 21-mer with an average GC content of 50%, fully complementary with its target (no mismatches), will have a priming efficiency number of approx 400. A similar 30 mer will have a PE of approx 500. When such primers contain mismatches with a given target, such that they will marginally hybridize and prime, the priming efficiency will be approx 200 (*see* Fig. 3).

Priming efficiency is obviously useful in accurately predicting false priming. It has also proven valuable in the selection of *consensus primers* and *unique primers*, and it has been effective in selecting sequencing primers from a library of previously synthesized oligonucleotides. This eliminates the time and expense of a new synthesis.

Sequencing Primer and Hybridization Probe Selection

Along with its search for PCR primers, OLIGO includes comprehensive searches for DNA sequencing primers and hybridization probes. These searches are also initiated from the *Search for Primers & Probes* window.

When the sequencing primer search is chosen, the same subsearch group is selected as for PCR. The **Frequent Oligos** subsearch is not required for optimal sequencing primer performance. Although the subsearches used are almost the same as those for the PCR primer search, the threshold values are different in certain respects. First, there are advantages to using a sequencing primer with a very high Tm, so unlike PCR, the sequencing primer search selects high Tm oligos, including those with the highest Tms. The rest of the parameters are set close to those used for PCR, because the hybridization and priming requirements for both are very similar. After selecting the **Sequencing Primers** search, the user will set search stringency in the *Parameters* window as in the PCR search. The user will also check the **Complex Substrate** box (usual set off for sequencing in plasmids) and which strand (+ or –) to select the primer.

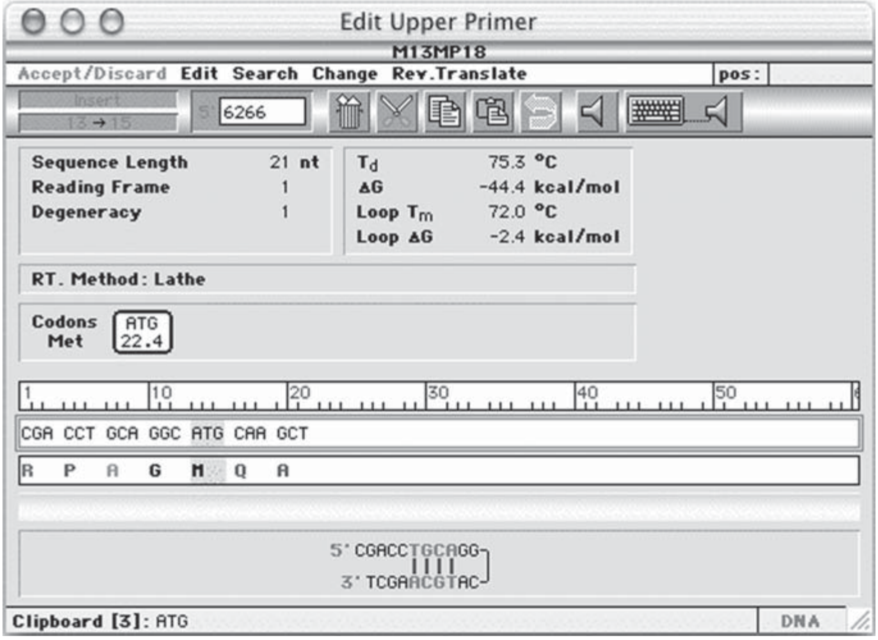
Once the search is initiated, it proceeds as it does with the PCR search. But the chosen (optimal) sequencing primers are displayed in a *Selected Oligos* window. This window displays the oligonucleotide number, position on the template, Tm, 3' terminal stability (specificity), and GC clamp stability. The user can sort by any one of these columns of data and, thereby further refine the primer selection to his or her liking. Sequencing primers can be exported to an oligonuclotide database and order form as can PCR primers.

The hybridization probe search is similar to the PCR and sequencing primer search, except that those subsearches that are designed to optimize the priming process are turned off, and those designed to enhance hybridization are turned on (and turned up). The probe search includes the **GC Clamp**, the **Stability**, the **Homo-oligomer/Sequence Repeats**, and the **Hairpin-free** subsearches. Once the probe search is complete, optimal probes are, again, listed in the *Selected Oligos* window. When a particular probe is selected, it is highlighted as the upper (or lower) primer on the OLIGO melting temperature screen. Along with standard probe selection, OLIGO can also be used to select optimal TaqMan probes and primers for this popular application.

Mutagenesis Primer Design

Site-specific mutagenesis is a very popular technique in the molecular biology laboratory. Perhaps the easiest way to insert a specific mutation into a gene is using primers containing mutagenic oligonuceotides. OLIGO 6 includes a mutagenesis feature that simplifies the design of the mutagenesis experiment. The *Mutagenesis* window displays a single line of 50–100+ bases at a time for viewing or editing, depending on the computer resolution. The user can switch from a nucleic acid sequence to a protein sequence in this edit mode by re-positioning the cursor. This window also displays the most stable hairpin, codon probabilities, and several other characteristics of the sequence displayed (*see* Fig. 4).

In addition to the Toolbar information, the window displays: sequence length, active reading frame, degeneracy, Tm, G, hairpin loop Tm, and G, reverse translation method, codon frequency for a given organism, nucleotide sequence with position numbers, translated protein sequence, the strongest hairpin loop of the displayed



Sequence Length 21 nt
Reading Frame 1
Degeneracy 1

T_d 75.3 °C
ΔG -44.4 kcal/mol
Loop T_m 72.0 °C
Loop ΔG -2.4 kcal/mol

RT. Method: Lathe

Codons Met ATG 22.4

```
1        10       20       30       40       50
CGA CCT GCA GGC ATG CAA GCT
R   P   A   G   M   Q   A

5' CGACCTGCAGG
        ||||
3' TCGAACGTAC
```

Clipboard [3]: ATG

Fig. 4. Mutagenesis Edit window.

sequence, and the size and contents of the Clipboard; the sequence to be inserted with the **Paste** command (*see* Fig. 4).

The various colors of residues in the amino acid sequence represent the codon frequencies of each residue according to the codon table selected in the **Change** submenu. The information displayed in the mutagenesis window, allow the user to efficiently design mutagenetic primers with confidence.

Consensus Primers and Unique Primers Selection

PCR has become a universal tool used by molecular biologists, and is now being used in a wide variety of creative ways. Among them are consensus primer PCR and unique primer PCR. The **Consensus Primers** option is useful when the researcher wishes to select and synthesize a primer pair that can be used to amplify a number of homologous genes. Again, selecting a consensus primer pair would be difficult without the priming efficiency algorithm.

To begin the consensus primer selection function in OLIGO 6, the user sets up the standard PCR search, setting the search stringency and the minimum priming efficiency for the consensus primers in the parameters window. The files are then selected from which the consensus primers are to be selected, by clicking the **Consensus Primers** button on the main search window and then adding the files using the *Select Files* window. The search proceeds in the manner of the standard PCR search. The primer pairs displayed in the primer pairs window following the completion of the search will, by definition, prime in each of the consensus sequences with a priming efficiency at or above the threshold value set in the parameters window.

Unique primer selection is the reverse of the consensus primer-selection process. Here, the user wishes to selectively prime and amplify one particular gene in a PCR reaction from a number of homologous genes. The unique primer selection search can also be useful to differentiate between closely related bacterial or viral strains, or any group of closely related organisms. To initiate unique primer selection, the **Continue <False Priming> Search in Other Files** subsearch is selected instead of the **Consensus Primer** button. Here, it is the priming efficiency threshold for false priming (set in the parameters window) that determines whether a particular primer is selected as a *unique* primer.

Peripheral and Support Functions

In addition to the core features of the OLIGO program described earlier, a number of support features are included. For example, OLIGO 6, includes a search for restriction enzymes, with a cut table and map, a search for a sequence string (including wobbles and termination codons), a search for palindromes and a search for stems of hairpin loops. This specialized hairpin loop search can play a helpful role in the design of synthetic genes. It also has a number of functions that have been added to work with protein sequences. Protein sequences can be loaded into the oligo program, reverse translated (four different methods of reverse translation are provided, including degenerate and codon table methods) with oligo degeneracies displayed, and the protein sequence searched for potential restriction sites.

Under the **Analyze** menu, hybridization time can be calculated and the concentrations and volumes of the primers, PCR product, or the entire sequence can be quickly determined. The internal stability (free energy of pentamer subsequences) of an oligo can also be analyzed, assisting the user in choosing the ideal internal design of a primer-relatively unstable on the 3' end and stable elsewhere. DNA and protein sequence files from a number of formats open easily in OLIGO, including GenBank and PIR files, FASTA, and FASTN along with files from word processors saved as text files.

Suggested Readings

Oligo

Rychlik, W. and Rhoads, R. E. (1989) A computer program for choosing optimal oligonucleotides for filter hybridization, sequencing and in vitro amplification of DNA, Nucleic Acids Res. 17, 8543–8551.

Rychlik, W. (1999) Oligo Primer Analysis Software, v. 6, Molecular Biology Insights, Cascade, CO.

PCR Primer Search and Selection

Breslauer, K. J., Frank, R., Blocker, H., and Markey, L. A. (1986) Predicting DNA duplex stability from the base sequence, Proc. Natl. Acad. Sci. USA 83, 3746–3750.

Freier, S. M., Kierzek, R., Jaeger, J. A., Sugimoto, N., Caruthers, M. H., Neilson, T., and Turner, D. H. (1986) Improved free-energy parameters for predictions of RNA duplex stability, Proc. Natl. Acad. Sci. USA 83, 9373–9377.

Rychlik, W. (1995) Priming Efficiency in Polymerase Chain Reaction, Biotechniques 18, 84–90.

Priming Efficiency Measurements for Priming and False Priming

Steffens, D. L., Sutter, S. L., and Roemer, S. C. (1993) An alternate universal forward primer for improved automated DNA sequencing of M13, BioTechniques 15, 580–582.

22 Statistical Modeling of DNA Sequences and Patterns

Gautam B. Singh

Introduction

Of fundamental importance in bioinformatics are the mathematical models designed for modeling the biological sequence and to use that as the basis for detection of patterns. Patterns at the various levels of abstractions are the drivers of genomics and proteomics research. Starting at the fine level of granularity, the patterns are comprised of the splice sites, binding sites, and domains. These are subsequently utilized for the definition of patterns at a higher level of abstraction such as introns, exons, repetitive DNA, and locus-control regions.

DNA sequences are often modeled as probabilistic phenomena, with the patterns of interest being defined as samples drawn from the underlying random process. For example, the underlying DNA sequence is modeled as a Markov chain of random variables taking on the values (A, C, T, G). Given this underlying assumption, one may then model a splice site as a function P that assigns a sample sequence, S, of 8–10 nucleotides a value equivalent to their probability of being a splice site.

A pattern-detection algorithm would consider each substring from a DNA sequence that could potentially be a splice site and assign a probability to each candidate. The substrings scoring a high value may be further investigated using the appropriate wet-bench approaches. Although this is a simple illustration, it does bring forth an important point. The underlying models that define the DNA sequences and their accuracy are ultimately a determinant of the accuracy with which the patterns are subsequently detected. The modeling of the sequences and that of patterns are two complementary objectives. Sequence models provide a basis for establishing the significance of patterns observed, while the pattern models help us look for specific motifs that are of functional significance. Therefore we must consider both of these issues.

Sequence Models

The two main sequence models are the Independent Identically Distributed and Markov Chain models. Sequence models are needed to represent the background stochastic processes in a manner that enables one to analytically justify the significance of the observation. To provide an analogy, determinng the sequence model is similar to determining the probability of obtaining a *head* (H) while tossing a coin. (For a fair coin, this probability would be ½). In general however, we may estimate this probability by studying the strings of the heads and tail sequences that a given coin has produced in the past. Similarly, given the DNA sequence(s), we may induce the underlying model that represents the maximally likely automaton that produced the sequence.

Let us continue our analogy further. After the coin model has been induced, it would be possible to predict the probability of observing coin tossing pattern such as "three heads in a row," etc. Similarly, after inducing a DNA sequence model, it would be possible to deduce the expected frequency of occurrence of a DNA sequence pattern. This is helpful in classifying patterns in terms of their relative abundance in a sequence specific manner.

Independent Identically Distributed (IID)

The simplest of all the sequence models is the Independent Identically Distributed or IID model. In this model, each of the four nucleotides is considered to occur independent of each other. Furthermore, the probability of the occurrence of a given nucleotide at a given location is identical to the probability of its occurrence at another location. Thus, for example, assume that the sequence is defined using an IID random variable. It can take on the possible values defined by the DNA alphabet $\Sigma = (A, C, T, G)$. In this case, defining the individual probability values (p_A, p_C, p_T, and p_G) specifies the complete model for the sequence. The values may in turn be computed simply by considering the prevalence of each base in the given sequence. In statistical terms, the maximally likely or ML estimator for probability of occurrence of a given base is X is simply $\frac{n_x}{L}$ where n_x is the frequency of occurrence of the base X in a sequence of length L.

In general, the maximal likely estimator for the parameters may be used. Using the ML estimation, the probability of each base α, may be estimated as shown in Equation 1:

$$\hat{P}(\alpha) = \frac{n_\alpha(L)}{|L|} \qquad \text{[Eq. 1]}$$

This simply counts the relative frequency of nucleotide α in a sequence of length L. This estimator has the advantage of simplicity and usually works well when $|L|$ is large. It may not work well when $|L|$ is small.

Given the Model $\mathrm{M}_{\mathrm{IID}}$ has been induced from the sequence data, the probability of an occurrence of a pattern x may be computed as shown in Equation 2.

$$P(x \mid M_{IID}) = \prod_{i=1,\ldots,n(x)} P(x_i) \qquad \text{[Eq. 2]}$$

where $P\ (x_i)$ is the probability of nucleotide x_i at position i along the pattern. The model assumes that the parameters (probability of each of the four nucleotides) are independent of the position along the pattern.

Example 1:

Consider the following DNA Sequence of Length = 25.

SEQ = AACGT CTCTA TCATG CCAGG ATCTG

In this case the IID model parameters are $\left(\frac{6}{25}, \frac{7}{25}, \frac{7}{25}, \frac{5}{25}\right)$. This corresponds to the maximally likely estimation of the occurrence of each of the four bases given the alphabet S = (A, C, T, G). These are thus the IID parameters for the background sequence. The probability of finding the pattern CAAT on this sequence would be equal to $p_C.p_A.p_A.p_T$ or $\left(\frac{7}{25}\right)\cdot\left(\frac{6}{25}\right)\cdot\left(\frac{6}{25}\right)\cdot\left(\frac{7}{25}\right) = 0.0045$.

Markov Chain Models

In a Markov chain the value taken by a random variable is dependent upon the value(s) taken by the random variable in a previous state(s). The number of historical states that influence the value of the random variable at a given location along the sequence is also known as the degree of the Markov process. The first-degree Markov chain model has $|\Sigma| + |\Sigma|^2$ parameters, corresponding to the individual nucleotide frequencies as well as dinucleotide frequencies. In this manner, this model permits a position to be dependent on the previous position. However, the frequencies are modeled in a position-invariant manner and thus may not be suitable for modeling signals.

This sequence model M is defined on the sample space Σ^* and assigns a probability to every sequence x of length $n(x)$ on Σ^* (*see* Equation 3):

$$P(x \mid M) = P_1(x_1) \prod_{i=2,\ldots,n(x)} P_2(x_i|x_{i-1}) \qquad \text{[Eq. 3]}$$

where P_1 is a probability function on Σ that models the distribution of α's at the first position in the sequence and P_2 is the conditional probability function on $\Sigma \times \Sigma$ that models the distribution of β's at position $i > 1$ on the alphabet symbol α at position i–1.

The parameter estimation using the *Maximally Likely* estimator proceeds in a manner analogous to the IID model estimation. The transition probabilities are however estimated using Bayes theorem as shown in Equation 4:

$$P_2(\beta|\alpha) = \frac{P(\alpha\beta)}{P(\alpha)} \qquad \text{[Eq. 4]}$$

In this manner, the conditional transitional probabilities of finding a base β at position (i) given that the base α was found at position (i–1) is computed by finding the abundance of the dinucleotide $\alpha\beta$ as a fraction of the abundance of the nucleotide α.

Example 2:

Consider once again the same 25-Nucleotide Sequence as shown earlier.

SEQ = AACGT CTCTA TCATG CCAGG ATCTG

While considering the first-degree Markov chain models, the 4-parameters corresponding to individual nucleotide frequencies, and the 4^2 parameters corresponding to the dinucleotide frequencies need to be computed. The Σ parameters are the same as before = $\left(\frac{6}{25}, \frac{7}{25}, \frac{7}{25}, \frac{5}{25}\right)$.

In order to compute P_2, the $\Sigma \times \Sigma$ conditional probability values, the dinucleotide frequencies and probabilities are computed from the sequence data. The dinucleotide frequencies and the probabilities are shown below (with the parenthesized numbers representing the probabilities):

freq (AA) = 1 $\left(\frac{1}{24}\right)$	freq (AC) = 1 $\left(\frac{1}{24}\right)$	freq (AT) = 3 $\left(\frac{3}{24}\right)$	freq (AG) = 1 $\left(\frac{1}{24}\right)$
freq (CA) = 2 $\left(\frac{2}{24}\right)$	freq (CC) = 1 $\left(\frac{1}{24}\right)$	freq (CT) = 3 $\left(\frac{3}{24}\right)$	freq (CG) = 1 $\left(\frac{1}{24}\right)$
freq (TA) = 1 $\left(\frac{1}{24}\right)$	freq (TC) = 4 $\left(\frac{4}{24}\right)$	freq (TT) = 0 $\left(\frac{0}{24}\right)$	freq (TG) = 2 $\left(\frac{1}{24}\right)$
freq (GA) = 1 $\left(\frac{1}{24}\right)$	freq (GC) = 1 $\left(\frac{1}{24}\right)$	freq (GT) = 1 $\left(\frac{1}{24}\right)$	freq (GG) = 1 $\left(\frac{1}{24}\right)$

Table 1
Conditional Nucleotide Probabilities for the 25-nt Example Sequence

↓ S_{i-1} S_i →	*A*	*C*	*T*	*G*
A	25/144	25/144	75/144	25/144
C	50/168	25/168	75/168	25/168
T	25/168	100/168	0	50/168
G	25/120	25/120	25/120	25/120

Table 2
Patterns and Probabilities Deduced Using IID and Markov Models

	Probability	
Pattern	*IID Model*	*Markov Model*
ATTTA	2.57×10^{-3}	2.0×10^{-3}
TGTTTTG	1.06×10^{-4}	1.64×10^{-4}
TTTTGGGG	1.40×10^{-5}	4.153×10^{-5}
CTTTTACCAAT	5.181×10^{-7}	7.596×10^{-7}
TCTTTATCTTTGCG	6.545×10^{-9}	1.968×10^{-9}
CTGAACATTGATGCA	1.238×10^{-9}	3.19×10^{-9}

The conditional probabilities are next computed using the Bayes theorem (*see* Equation 4). For example, the probability of finding "*C*" at position (i+1) given that an "*A*" has been found at position (i) is $P(C \mid A) = \frac{p_{AC}}{p_A} = \frac{1/24}{6/25} = \frac{25}{144}$. For large sequences the conditional probability $P(S_i|S_{i-1})$ approaches

$$\frac{freq(S_iS_{i-1})}{freq(S_{i-1})}$$

The conditional probabilities for the example sequence are shown in Table 1. Using these model parameters, the probability of finding the pattern CAAT in this sequence using the first order Markov model of the underlying sequence would be equal to $P(C)\cdot P(A|C)\cdot P(A|A)\cdot P(T|A)$ or $\left(\frac{7}{25}\right)\cdot\left(\frac{50}{168}\right)\cdot\left(\frac{25}{144}\right)\cdot\left(\frac{75}{144}\right) = 0.0075$. This is contrast to the ID sequence model probability of the pattern being 0.0045.

Example 3:

The sequence models described above were induced from a 76kb sequence from the beta-globin gene (L22754+HUMHBB). The pattern probabilities of the following patterns were computed for both sequence models. The patterns were carefully selected to be of varying lengths. A PERL program (*markov.pl*) for inducing sequence models and computing pattern probabilities is provided on the accompanying CD-ROM. Table 2 depicts the patterns as well as the probabilities of their occurrence deduced under the IID and Markov models.

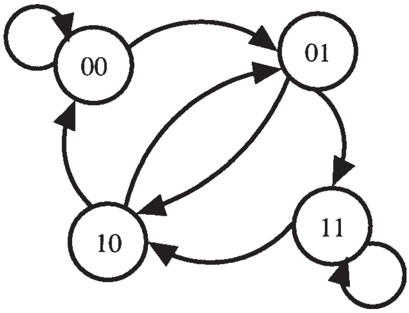


Fig. 1. The set of possibe transitions in a 2nd. Order Markov chain. Note that only 8 nonzero probabilities need to be estimated for this model, as not all of the 16 transitions are possible.

Higher-Order Markov Models

Higher-order Markov chains have been described. For example, the *nth* order Markov process has a memeory of n, and thus the occurrence of a nucleotide depends on the previous n nucloetides. The probability of observing a sequence x is defined in Equation 5, in a manner similar to the first order Markov chains.

$$P(x \mid M) = P_1(x_1) \prod_{i=2,\ldots,n(x)} P_2(x_i | x_{i-1},\ldots,x_{i-n}) \qquad \text{[Eq. 5]}$$

An *nth* order Markov chain over some alphabet A is equivalent to a first order Markov chain over the alphabet A^n of n-tuples. This follows from calculating the probability of A and B, given B is the probability of A given B, i.e., $P(x_k|x_{k-1},\ldots,x_{k-n}) = P(x_k,x_{k-1},\ldots,x_{k-n+1}|x_{k-1},\ldots,x_{k-n})$. That is, the probability of x_k given the n-tuple ending in x_{k-1} is equal to the probability of the n-tuple ending in x_k given the n-tuple ends in x_{k-1}.

Consider the simple example of a second order Markov chain for sequences of only two different characters "0" and "1". A sequence is translated to a sequence of pairs, so the sequence 011101 becomes 01-11-11-10-01. All transitions are not possible in this model, as only two different pairs can follow a given letter. For example, the state 01 can only be followed by the states 10 and 11, and not by the state 00 or 01. The set of possible transitions are shown in Fig. 1.

A second order model for DNA is equivalent to a first order model over the Σ^2 alphabet that is comprised of the 16 dinucleotedes. A sequence of five bases, CGACG would be treated as a chain of four states, CG-GA-AC-CG, with each symbol being drawn from the Σ^2 alphabet. Nonetheless, the framework of high order models is often preferable for analysis of higher order stochastic sequences.

Pattern Models

In the previous section our goal was to characterize the "sea" of data in which these biological "nuggets" of information are hidden. In contrast, our goal in this section is to model the nuggets themselves. Thus, this section describes the statistical modeling procedures for DNA patterns. We now focus on modeling motifs that are associated with certain biological functions.

Table 3
Weight Matrix for TATAA Box

T	6	49	1	56	6	22	6	20
C	14	6	0	0	3	0	1	2
A	8	4	58	4	51	38	53	30
G	32	1	1	0	0	0	0	8

There are a growing number of well-established patterns that we may wish to model and search for in DNA sequences. Often these patterns of functional significance are brought forth after an alignment of sequences belonging to a particular family. Such a multiple sequence alignment is often interspersed with gaps of varying sizes. However, there are sections in the final alignment that are free of gaps in all of the sequences. These fixed-size, ungapped, aligned regions represent the types of patterns that are modeled to permit their identification in an anonymous segment of DNA. The statistical techniques that may be employed for developing such a closed-form representation of a set of patterns are described in the following.

Weight Matrices

A DNA sequence matrix is a set of fixed-length DNA sequence segments aligned with respect to an experimentally determined biologically significant site. The columns of a DNA sequence matrix are numbered with respect to the biological site, usually starting with a negative number. A DNA sequence motif can be defined as a matrix of depth 4 utilizing a *cut-off* value. The 4-column/mononucleotide matrix description of a genetic signal is based on the assumptions that the motif is of fixed length, and that each nucleotide is independently recognized by a *trans*-acting mechanism. For example, the frequency matrix has been reported for the TATAA box (*see* Table 3).

If a set of aligned signal sequences of length L corresponds to the functional signal under consideration, then $F = [f_{bi}]$, $(b \in \Sigma)$,$(j = 1..\ L)$ is the nucleotide frequency matrix, where f_{bi} is the absolute frequency of occurrence of the *b-th* type of the nucleotide out of the set $\Sigma = (A, C, G, T)$ at the *i-th* position along the functional site.

The frequency matrix may be utilized for developing an ungapped score model when searching for sites in a sequence. Typically a log-odds scoring scheme is utilized for purpose of searching for pattern x of length L as shown in Equation 6. The quantity $e_i(b)$ specifies the probability of observing base b at position i. It is defined using the frequency matrix such as the one shown above. The quantity $q(b)$ represents the background probability for the base b.

$$S = \sum_{1=1}^{L} \log \frac{e_i\,(x_i)}{q\,(x_i)} \qquad \text{[Eq. 6]}$$

The elements of

$$\log \frac{e_i\,(x_i)}{q\,(x_i)}$$

behave like a scoring matrix similar to the PAM and BLOSUM matrices. The term Position Specific Scoring Matrix (PSSM) is often used to define the pattern search with matrix. A PSSM can be used to search for a match in a longer sequence by evaluating a score S_j, for each starting point j in the sequence from position 1 to $(N–L+1)$ where L is the length of the PSSM.

A method for converting the frequency matrix into a weight matrix has been proposed by Bucher (1990). The weights at a given position are proportional to the logarithm of the observed base frequencies. These are increased by a small term that prevents the logarithm of zero and minimizes sampling errors. The weight matrix is computed as shown in Equation 7. The term e_{bi} represents the expected frequency of base b at position i, c_i a column specific constant, and s, a smoothing percentage.

$$W(b,i) = \ln\left(\frac{f_{bi}}{e_{bi}} + \frac{s}{100}\right) + c_i \qquad \text{[Eq. 7]}$$

These optimized weight matrices can be used to search for functional signals in the nucleotide sequences. Any nucleotide fragment of length L is analyzed and tested for assignment to the proper functional signal. A matching score of

$$\sum_{i=1}^{L} W\left(b_i,i\right)$$

is assigned to the nucleotide position being examined along the sequence. In the search formulation, b_i is the base at position i along the oligonuclotide sequence, and $W(b_i,i)$ represents the corresponding weight-matrix entry for base b_i occurring along the ith. position in the motif.

Profiles are similarly defined for modeling functional motifs in amino-acid sequences. A profile is a scoring matrix $M(p,a)$ comprised of 21 columns and N rows, where N is the length of the motif. The first 20 scores represent the weight for each individual amino acid, and the 21st column specifys the cost associated with an insertion or deletion at that position. The value of the profile for amino acid "*a*" defined for position p is

$$M\left(a,p\right) = \sum_{b=1}^{20} W\left(b,p\right) \times Y(b,a)$$

where $Y(b,a)$ is Dayhoff's matrix and $W(b,p)$ is the weight for the appearance of amino acid b at position p. The position specific weight is defined by $\log(f[b,p]/N)$, or the frequency of occurrence of the amino acid as b as a fraction of the total N sequences utilized for construction of the profile, with a frequency of 1 being used for any amino acid that does not appear at position p.

Position-Dependent Markov Models

Markov models have been considered as a means to define the background DNA sequence. This model enabled us to define the probability of a nucleotide conditioned upon the nucleotides occurring in the previous position. However the modeled dependency is position-invariant. A position-dependent Markov model may be utilized for the representation of a sequence signal or motif. This model is defined on the sample space Σ^n and assigns a probability to every sequence x on Σ^n (*see* Equation 8):

$$P(x \mid M) = P_1(x_1) \prod_{i=2,\ldots,n} P_{2,i}(x_i|x_{i-1}) \qquad \text{[Eq. 8]}$$

This model has $|\Sigma| + (n–1)*|\Sigma|^2$ parameters. This model permits position-specific dependencies on the previous position by allowing the association of a unique set of transition probabilities with each position along the signal. This model assumes that sufficient training data is available to induce position specific Markov probabilities.

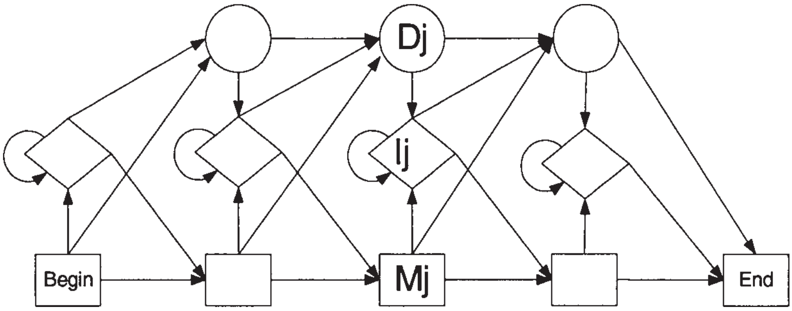


Fig. 2. A profile HMM utilizes the insert (diamond) and delete (circle) states. The delete states are *silent* and are not associated with the emissions of any symbols. The parameters of a profile HMM are learned from a multiple sequence alignment.

Hidden Markov Models

There are several extensions to the classical Markov chains. The Hidden Markov models (HMM) are one such extension. The rationale for building a HMM comes from the observation that as we search a sequence, our observations could arise from a model characterizing a pattern, or from a model that characterizes the background. Hidden Markov DNA sequence models are developed to characterize a model as an *island* within the sea of *nonisland* DNA. The Markov chain characterizing the *island* and *nonisland* need to be present within the same model, with the ability to switch from one chain to the other. In this manner, a HMM utilizes a set of hidden states with an emission of the symbols associated with each state.

From a symbol-generation perspective, the state sequence executed by the model is not observed. Thus, the state sequence must be estimated from the observed symbols generated by the model. From a mathematical perspective, the HMM is characterized by the following parameters:

- Σ is an alphabet of symbols.
- Q is a set of states that emit symbols from the alphabet Σ.
- A = (a_{kl}) is $|Q|\times|Q|$ matrix of state transition probabilities.
- E = $(e_k[b])$ is a $|Q|\times|\Sigma|$ matrix of emission probabilities.

Although a general topology of a fully connected HMM allows state transitions from any state to any other, this structure is almost never used. This is primarily due to the inadequacy of the available data for training a model with the large number of parameters needed for a fully connected HMM for any practical problem. Often, the over-generalized model produces sub-optimal results due to the lack of training data. Consequently, more restrictive HMMs that rely on the problem characteristics to suitably reduce the model complexity and the number of model parameters that are needed are utilized. One such model is defined to be the *profile-HMM*, which is induced from a multiple sequence alignment. The structure of a profile HMM is shown in Fig. 2.

The parameters of a profile HMM are estimated using the sample alignments of the sequences used for training. The transitions and the emissions of the symbols in the alignment are used to derive the Maximally Likely (ML) estimator of the HMM parameters. These values are assigned as shown in Equation 9. The actual transition and emission frequencies A_{kl} and $E_k(a)$ respectively, are used to define the transition and emission probabilities, A_{kl} and $E_k(a)$. Furthermore, pseudo-counts are added to the observed frequencies to avoid zero probabilities. The simplest pseudo-count method is Laplace's rule, which requires adding one to each frequency.

$$a_{kl} = \frac{A_{kl}}{\sum_X A_{kX}} \quad \text{and} \quad e_k(a) = \frac{E_k(a)}{\sum_Y E_k(Y)} \qquad \text{[Eq. 9]}$$

Example 4:

Consider the following multiple sequence alignment defined on amino acid residues for five globin sequences.

```
HBA_HUMAN          ... VGA--HAGEY ...
HBB_HUMAN          ... V----NVDEV ...
GLB3_CHITP         ... VKG------D ...
LGB2_LUPLU         ... FNA--NIPKH ...
GLB1_GLYDI         ... IAGADNGAGV ...
Match States           ***  *****
```

A HMM with 8 match states may be constructed based on this alignment. The residues **AD** in **GLB1_GLYD1** are treated as insertions, with respect to the consensus. In match state 1, the emission probabilities are (using Laplace's rule):

$$e_{M1}(V) = \frac{4}{25},\ e_{M1}(F) = \frac{2}{25},\ e_{M1}(I) = \frac{2}{25},\ \text{and } e_{M1}(a) = \frac{1}{25} \text{ for all others}$$

The transition probabilities from match state 1 are as follows:

$$a_{M1,M2} = \frac{5}{8},\ a_{M1,D2} = \frac{2}{8},\ a_{M1,I1} = \frac{1}{8}\ ,$$

corresponding to the one deletion in **HBB_HUMAN**, and no insertions. The emission probabilities for the state I_1 will be all equal to (1/20). The Viterbi algorithm yields the optimal path through the HMM, as well as the log-odds score for observing a given sequence using an HMM. It is commonly used to match a profile HMM to a sequence. The Viterbi algorithm is a recursively defined optimization procedure that is quite similar to the dynamic programming algorithm used for sequence alignment.

The various scores for matching a sequence to the profile HMM are defined in Equation 10. In this formulation, $V_j^M(i)$ represents the log-odds score of the best path matching subsequence $x_{1\ldots i}$ to the submodel up to state j, ending with x_i being emitted by state M_j. Similarly $V_j^I(i)$ is the score of the best path ending in x_i being emitted by I_j, and $V_j^D(i)$ for the best path ending in state D_j.

$$V_j^M(i) = \log\frac{e_{M_j}(x_i)}{q_{x_i}} + \max\begin{cases} V_{j-1}^M(i-1) + \log a_{M_{j-1}M_j} \\ V_{j-1}^I(i-1) + \log a_{I_{j-1}M_j} \\ V_{j-1}^D(i-1) + \log a_{D_{j-1}M_j} \end{cases}$$

$$V_j^I(i) = \log\frac{e_{I_j}(x_i)}{q_{x_i}} + \max\begin{cases} V_j^M(i-1) + \log a_{M_jM_j} \\ V_j^I(i-1) + \log a_{I_jM_j} \\ V_j^D(i-1) + \log a_{D_jM_j} \end{cases} \qquad \text{[Eq. 10]}$$

$$V_j^D(i) = \max\begin{cases} V_{j-1}^M(i) + \log a_{M_jD_j} \\ V_{j-1}^I(i) + \log a_{I_{j-1}D_j} \\ V_{j-1}^D(i) + \log a_{D_{j-1}D_j} \end{cases}$$

Generally, there is no emission score $e_{I_j}(x_i)$ in the equation for $V_j^I(i)$ as it is often assumed that the emission distribution from the insert states I_j is the same as the background distribution. Also, the $D \rightarrow I$ and $I \rightarrow D$ transition terms may not be present, and those transition probabilities may be very close to zero.

Mixture Models

Mixture models are defined in relation to the sample space Σ^n. The mixture model M is a mixture of component models M_i,i = 1,..., k, and assigns a probability to every sequence x on Σ^n defined in Equation 11.

$$P(x \mid M) = \prod_{i=1,\ldots,k} P(x \mid M_i)\, P\,(M_i) \qquad \text{[Eq. 11]}$$

where $P(M_i)$ is the weight of component model M_i in the mixture. Any probability model may be used as a component model, as the mixture model may have component models of a different type.

A mixture model is best suited for modeling data that is comprised of subgroups. In this manner, an observed data set may be assigned to a class if a high probability is assigned to it by at least one component model of sufficient weight. This may be considered to be a stochastic analog of a weighted OR function.

Consider for example a set D of short aligned sequences corresponding to some functional site. Further assume that we have a reason to believe that the observed sequences are characterized to belong to two categories. The goal to establish if an observed sequence x is similar to set D, may be achived by developing a mixture model for the set D. The mixture model is comprised of two constituent sub-models that are represented in the set.

Goodness of Fit

As described in the previous sections, there are often several methodologies for developing a model for a pattern. Consequently, the natural question to ask is: is there is a systematic methodology that may be utilized to evaluate which of the possible models is best suited for the data at hand? We may utilize the *goodness of fit* measure, described below, to estimate how well a given model represents the observed dataset. Generally, all data items d in the training, or another data set, D are considered to be independent.

Under the independence assumption, the likelihood that dataset D may be estimated as a product of all the probabilities of individual observations d is:

$$P(D \mid M) = \prod_{d \in D} P(d \mid M) \qquad \text{[Eq. 12]}$$

Rewriting Equation 11 in its log-likelihood form:

$$\log P(D \mid M) = \prod_{d \in D} \log P(d \mid M) \qquad \text{[Eq. 13]}$$

This is often desirable to prevent numeric underflows that are likely to occur when small numbers are multiplied. In this manner, one may choose a model instance M^* that best fits the dataset D according to the maximum likelihood principle defined in Equation 14.

$$M^* = \arg \max_M \; (P\,[D \mid M]) = \arg \max_M \; (P\,[D \mid M]) \qquad \text{[Eq. 14]}$$

Example Problems

The following examples and their solutions review the key concepts of this chapter.

1. Compute the background parameters for the following sequence:

```
ATTAG GCACG CATTA TAATG GGCAC CCGGA AATAA CCAGA GTTAC GGCCA
```

a. Assuming an IID model? Answer:

```
IID parameters

A : 17 (0.34)
B : 12 (0.24)
C : 12 (0.24)
T : 9 (0.18)
```

b. Assuming a 1st Order Markov model? Answer:

```
1-st Order Markov Probabilities Dinucleotide Frequencies

AA  4  (0.0816326530612245)
CA  5  (0.102040816326531)
AC  4  (0.0816326530612245)
CC  4  (0.0816326530612245)
GA  2  (0.0408163265306122)
TT  3  (0.0612244897959184)
AG  3  (0.0612244897959184)
GC  4  (0.0816326530612245)
CG  3  (0.0612244897959184)
GG  5  (0.102040816326531)
AT  5  (0.102040816326531)
TA  5  (0.102040816326531)
CT  0  (0)
TC  0  (0)
GT  1  (0.0204081632653061)
TG  1  (0.0204081632653061)

Prob X|Y is derived from Prob (XY)/Prob (Y)

Markov Probabilities (1-st Order)

A|C  5/12  (0.425170068027211)
C|A  4/17  (0.240096038415366)
C|C  4/12  (0.340136054421769)
A|G  2/12  (0.170068027210884)
T|T  3/9   (0.340136054421769)
G|A  3/17  (0.180072028811525)
C|G  4/12  (0.340136054421769)
G|C  3/12  (0.255102040816327)
G|G  5/12  (0.425170068027211)
A|T  5/9   (0.566893424036281)
T|A  5/17  (0.300120048019208)
C|T  0/9   (0)
T|C  0/12  (0)
T|G  1/12  (0.0850340136054422)
G|T  1/9   (0.113378684807256)
A|A  4/17  (0.240096038415366)
```

c. How many parameters would need to be estimated if a 2nd Order Markov model is assumed for the background sequence? Answer:

```
A second order Markov Process for modeling a sequence will have
the following transitions:

AA|AA AT|AA AC|AA AG|AA
CA|CC CC|CC CT|CC CG|CC
```

```
TA|TT TC|TT TT|TT TG|TT
GA|GG GC|GG GT|GG GG|GG
TA|AT TC|AT TT|AT TG|AT
CA|AC CC|AC CT|AC CG|AC
GA|AG GC|AG GT|AG GG|AG
....
....
```

64 transition probabilities will be defined, as there are only 4 valid transitions from any of the 16 states. (Note that we would have 16x6 = 256 possible transitions if all transitions were valid. However, transitions such as AT|CG, etc. are not valid).

2. Compute the match scores (defined in Equation 5) generated by searching the above sequence using the weight matrix described for TATAA-box described in Table 1. Assume IID background model.

```
Pattern Length = 8
zero th order Markov model for the sequence
is used for the bakground computation.

A : 17 (0.34)
C : 12 (0.24)
T : 9  (0.18)
G : 12 (0.24)
-------------------------------------------
zeros in the score matrix were changed to 1
as Eq. 5 requires taking a log (e-i(x-i)/q(x-i)).

Scores as computed using Equation (5):
Score[i] represents the score for pattern starting at position
"i".
Score[O] = -11.164
Score[l] =  -7.167
Score[2] = -11.868
Score[3] = -15.445
Score[4] = -12.798
Score[5] =  -5.578
Score[6] =  -8.025
Score[7] =  -5.836
Score[8] =  -5.412
Score[9] =   2.331
Score[10] =   0.462
Score[ll] =  -1.943
Score[12] =   3.863
Score[13] =  -7.968
Score[14] =  -6.633
Score[15] =  -9.617
Score[16] = -15.343
Score[17] = -11.227
Score[18] = -14.293
Score[19] = -13.711
Score[20] = -13.771
Score[21] =  -9.305
Score[22] = -13.624
Score[23] = -11.255
Score[24] =  -7.058
Score[25] =  -3.244
Score[26] =  -5.771
Score[27] =   0.400
Score[28] =  -2.515
```

```
Score[29] =  -3.093
Score[30] = -10.722
Score[31] =  -3.992
Score[32] =  -7.183
Score[33] =  -9.399
Score[34] = -10.319
Score[35] =  -4.364
Score[36] =  -8.227
Score[37] =  -6.236
Score[38] =  -5.188
Score[39] =  -9.458
Score[40] = -10.690
Score[41] = -10.789

Maximum Score at Location 12.
Maximum Score = 3.863(12)
```

3. Use the HMMER program (URL provided in Subheading “Suggested Reading”) to compute profile HMM for the following alignment.

```
A    C    G    A    C    G    A    C    G    A    C    G    .
.    .    G    G    G    A    A    A    G    G    .    G    A
ACG..AAATTT.A
```

   a. Apply the Viterbi algorithm to compute the probability of observing the sequence “AATTAATTAA” from this model.

```
Step 1. Sequences were represented in Stockholm alignment format.

           #STOCKHOLM 1.0
           seq1 ACGACGACGACG.
           seq1 ..GGGAAAGG.GA
           seq1 ACG..AAATTT.A

Step 2. Program hmmbuild was run. This produced the following
        alignment HMM

   HMMER 2.0 [2. 2g]
   NAME  prb
   LENG  13
   ALPH  Nucleic
   RF       no
   CS       no
   MAP   yes
   COM   hmmbuild prb.hmm prb.slx
   NSEQ  3
   DATE  Fri Oct 12 14:38:44 2001
   CKSUM 7915
   XT      -9967     -1 -1000 -1000  -9967    -1 -9967    -1
   NULT      -1  -9967
   NULE       0      0      0      0
   HMM        A      C      G      T
           m->m   m->i  m->d  i->m   i->i  d->m  d->d  b->m  m->e
           -412      * -2011
        1  1007   -592  -592  -592      1
        -     0      0     0     0
        -   -21  -6687 -7729  -894  -1115  -701  -1378  -412     *
        2  -592   1007  -592  -592      2
        -     0      0     0     0
        -   -21  -6687 -7729  -894  -1115  -701  -1378     *     *
        3  -807   -807  1193  -807      3
        -     0      0     0     0
        -  -442  -7108 -1962  -894  -1115  -701  -1378     *     *
```

```
     4   450   -592   387  -592     4
     -     0      0     0     0
     -   -21  -6687 -7729  -894  -1115 -1435   -666     *     *
     5  -592    450   387  -592     5
     -     0      0     0     0
     -   -21  -6687 -7729  -894  -1115  -380  -2112     *     *
     6   749   -807   234  -807     6
     -     0      0     0     0
     -   -16  -7108 -8150  -894  -1115  -701  -1378     *     *
     7  1193   -807  -807  -807     7
     -     0      0     0     0
     -   -16  -7108 -8150  -894  -1115  -701  -1378     *     *
     8   749    234  -807  -807     8
     -     0      0     0     0
     -   -16  -7108 -8150  -894  -1115  -701  -1378     *     *
     9  -807   -807   792   171     9
     -     0      0     0     0
     -   -16  -7108 -8150  -894  -1115  -701  -1378     *     *
    10   234   -807   171   171    10
     -     0      0     0     0
     -  -442  -7108 -1962  -894  -1115  -701  -1378     *     *
    11  -592    450  -592   387    11
     -     0      0     0     0
     -  -629  -6687 -1541  -894  -1115  -380  -2112     *     *
    12  -592   -592  1007  -592    12
     -     0      0     0     0
     -   -33  -6021 -7064  -894  -1115  -380  -2112     *     *
    13   986   -571  -571  -571    13
     -     *      *     *     *
     -     *      *     *     *      *     *      *     *     0
//
```

b. Validate your results by comparing it to probability values generated by HMMER package.

```
Step 3. To evaluate tbe probability of matching a sequence to an
        profile HMM, we can run the program hmmpfam. This produced
        the following result for AATTAATTAA. The match score was
        -15.4 (E-value of 2, with anything less than 10 being
        considered significant match).
```

```
hmmpfam - search one or more sequences against HMM database
HMMER 2.2g (August 2001)
Copyright (C) 1992-2001 HHMI/Washington University School of Medi-
cine
Freely distributed under the GNU General Public License (GPL)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
HMM file:          prb.hmm
Sequence file:            prb.seq
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
-
Query sequence: Query
Accession:      [none]
Description:    Sequence

Scores for sequence family classification score includes all do-
mains):
Model    Description                        Score    E-value  N
-------  -----------                        -----    ------- ---
prb                                         -15.4          2   1
```

```
Parsed for domains:
    Model    Domain  seq-f seq-t    hmm-f hmm-t     score  E-value
    -------- ------- ----- -----    ----- -----     -----  -------
    prb        1/1       1    10 []     1    13 []  -15.4        2

    Alignments of top-scoring domains:
    prb: domain 1 of 1, from 1 to 10: score -15.4, E = 2
                       *->acgacaaagacga<-*
                          a     aa++  a
           Query     1    AAT--TAATTA-A    10

    //
```

Glossary and Abbreviations

Alternative Hypothesis The alternative hypothesis, H1, is the conclusion that a statistical hypothesis test tries to establish. For example, in a new drug trial, the alternative hypothesis might be that the new drug has a different effect, on average, compared to that of the current drug.

Bayes' Theorem Bayes' Theorem allows new information to be used to update the conditional probability of an event.

Binomial Distribution A binomial random variable is typically used to model the number of successes in a series of trials, for example, the number of 'heads' occurring when a coin is tossed 100 times.

Conditional Probability The process of estimating the probability given the state of the environment. For example, the probability of an event E may be P_1 under one environment C_1, written as $P(E|C_1) = P_1$. While, the probability of the same event E may be P_2 under a different environment C_2, denoted as $P(E|C_2) = P_2$. In many situations, once more information becomes available, we are able to revise our estimates for the probability of further outcomes or events happening.

Continuous Random Variable A continuous random variable is characterized by its ability to take on an infinite number of possible values. Continuous random variables are usually measurements.

Cumulative Distribution Function All random variables (discrete and continuous) have a cumulative distribution function. It is a function giving the probability that the random variable X is less than or equal to x, for a given x.

Discrete Random Variable A discrete random variable may only take on a countable number of distinct values such as 0,1,2,3,... If a random variable can take only a finite number of distinct values, then it is discrete by definition. Discrete random variables are usually (but not always) counts.

Event An event is any collection of outcomes of an experiment.

Expected Value The expected value (or population mean) of a random variable indicates its average or central value. It is a useful summary value (a number) of the variable's distribution.

Independent Events Two events are independent if the occurrence of one of the events gives us no information about whether or not the other event will occur; that is, the events have no influence on each other.

Kolmogorov-Smirnov Test A test is used to establish if the sample of data is consistent with a specified distribution function. Alternatively, used to establish if two samples of data come from the same distribution.

Mutually Exclusive Events Two events are considered mutually exclusive (or disjoint) if it is impossible for them to occur together.

Normal Distribution Normal distributions is a probability distribution function of a class of continuous random variables. A normal random variable is capable of assuming any value on the real line.

Null Hypothesis The null hypothesis, H0, represents the default theory that is believed to be true or is the basis for argument. For example, in clinical trials, the null hypothesis might be that the new drug is no better, on average, than the current drug. In this case, we would write H0: there is no difference between the two drugs on average.

Poisson Distribution Poisson distribution is a probability distribution function for a class of discrete random variables. The corresponding Poisson random variable is the count of the number of events that occur in a certain time interval or spatial area. For example, the number of patterns found in a fixed size widows along the biological sequence.

Probability A probability provides a quantitative estimation of how likely or unlikely is the occurrence of a particular event. Probability is conventionally expressed on a scale from 0 to 1.

Probability Density Function Is defined for a continuous random variable. It may be integrated to obtain the probability that the random variable takes a value in a given interval.

Probability Distribution Is defined for discrete random variable. It is a list of probabilities associated with each of its possible outcome. It is also called the probability function or the probability mass function.

***p*-Value** The probability value (*p*-value) of a statistical hypothesis test is the probability of getting a value of the test statistic as extreme as or more extreme than that observed by chance alone, if the null hypothesis H0, is true.

Random Variable The outcome of an experiment need not be a number. For example, the outcome coin-toss is ‘heads’ or ‘tails’. We often want to represent the outcomes as numbers. A random variable is a function that associates a unique numerical value with every outcome.

Significance Level The significance level of a statistical hypothesis test is a fixed probability of wrongly rejecting the null hypothesis H0, if it is in fact true.

Subjective Probability A subjective probability of an individual’s estimation of how likely is the occurrence of a particular event. It is not based on any precise computation - simply a reasonable assessment by a knowledgeable person.

Test Statistic A test statistic is a quantity calculated from our sample of data. Its value is used to decide whether or not the null hypothesis should be rejected in our hypothesis test.

Variance The (population) variance of a random variable is a non-negative number providing an estimation of how widely spread the values of the random variable are likely to be. The larger the value of variance, the more scattered the observations.

Suggested Readings

Hidden Markov Models

Haussler, D., Krogh, A., Mian, I. S. and Sjolander, K. Protein modeling using hidden Markov models: analysis of globins, in: Proceedings of 26th Annual Hawaii

International Conference on System Sciences, IEEE Computer Society Press, vol. 1, pp. 792–802..

Krogh, A. An introduction to hidden Markov models for biological sequences, in: Computational Biology: Pattern analysis and machine learning methods, (Salzberg, S., Searls, D., and Kasif, S., eds.), Elsevier Press.

Ohler, U., Harbeck, S., Niemann, H., Noth, E., and Reese, M. G. (1999) Interpolated Markov chains for eukaryotic promoter recognition, Bioinformatics 15, 362–369.

HMMER (*see* Website: http://www.genetics.wustl.edu/eddy)

HMMPro (*see* Website: http://www.netid.com)

SAM (*see* Website: http://www.cse.ucsc.edu/research/compbio/sam.html)

Sequence Alignment

Smith, T. and Waterman, M. (1981) Identification of common molecular subsequences, J. Mol. Biol. 147, 195–197.

Quantitative Methods

Besemer, J. and Borodovsky, M. (1999) Heuristic approach to deriving models for gene finding, Nucleic Acids Res. 27, 3911–3920.

Feller, W. (1971) An introduction to probability theory and applications, vol. II, John Wiley and Sons, New York, NY.

Hayes, W. S. and Borodovsky, M. (1998) How to interpret an anonymous bacterial genome: machine learning approach to gene identification, Genome Res. 8, 1154–1171.

Weight Matrix and Profiles

Bucher, P. (1990) Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences, J. Mol. Biol. 212, 563–578.

Gribskov, M. and Veretnik, S. (1996) Identification of sequence patterns with profile analysis, Methods Enzymol. 266,198–212.

Gribskov, M., McLachlan, A., and Eisenberg, D. (1987) Profile analysis: Detection of distantly related protein, Proc. Natl. Acad. Sci. USA 87, 4355–4358,

Staden R. (1988) Methods to define and locate patterns of motifs in sequences, Comp. Appl. Biosci. 4, 53–60.

Staden R. (1988) Methods for calculating the probabilities of finding patterns in sequences, Comp. Appl. Biosci. 5, 89–96.

Substitution Matrices

Dayhoff, M. A., Schwartz, R. M., and Orcutt, B. C. (1978) A model for evolutionary change in proteins, in: Atlas of Protein Sequence and Structure, pp. 345–352.

Henikoff, S. and Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. USA 89, 10,915–10,919.

23 Statistical Mining of the Matrix Attachment Regions in Genomic Sequences

Gautam B. Singh

Introduction

The functional role of a cell can be thought of as reflecting the partitioning of genes into compartments of those that can be expressed and those that cannot be expressed. Every cell contains the same genetic information and expresses the same subset of genes for basic cellular function. There is a second set of genes that are uniquely expressed in each cell type. The biological mechanism by which such differential gene expression is achieved and mediated is brought about by the presence of simple DNA sequence patterns that are predominantly found within the neighborhood of the gene locus. These simple sequence patterns are embedded within the majority of nuclear noncoding DNA (i.e., in direct contrast to the 2% of the genome that corresponds to the protein coding regions).

Noncoding DNA is comprised of special sequences of regulatory importance such as introns, promoters, enhancers, Matrix Association Regions (MARs), and repetitive elements. Many of these regions contain patterns that represent functional control points for cell-specific or differential gene expression. Others such as repetitive DNA sequence elements may serve as a biological clock. These and numerous other examples indicate that the patterns embedded in the eukaryote DNA may play a critical role for viability.

MARs: Background and Data Models

The Matrix or Scaffold Attachment regions are relatively short (100–1000 bp long) sequences that anchor the chromatin loops to the nuclear matrix. MARs are often associated with the origins of replication (ORI). They usually possess a concentrated area of transcription factor binding sites. Approximately 100,000 matrix attachment sites are believed to exist in the mammalian nucleus, where ~30,000–40,000 serve as ORIs. MARs have been observed to flank the ends of genic domains encompassing various transcriptional units. It has also been shown that MARs bring together the transcriptionally active regions of chromatin such that transcription is initiated in the region of the chromosome that coincides with the surface of nuclear matrix.

There are two potentiative states, i.e., open or closed, and two classes of potentiated open euchromatin, i.e., constitutive and tissue/cell-specific facultative. The constitutive class, e.g., those of the housekeeping genic domains, always maintain that segment of the genome in a transcriptionally favorable open chromatin conformation. In contrast, the tissue/cell-specific facultative class impose the open conformation onto a segment of the genome in a tissue/cell-specific manner. The two potentiative states, ready/open or off/closed, are correlated with the presence or absence of multiple factors interacting with element(s) that are far distal of their respective gene-specific promoters and enhancers at regions of locus control.

To date, three classes of elements that act as regions of locus control have been identified. This suggests that potentiation can be achieved by multiple and/or redundant means. The classes of elements are the Matrix Attachment Regions/Scaffold Attachment Regions (MARs/SARs); Specialized Chromatin Structures (SCSs); and Locus Control Regions (LCRs). These elements provide a dominant chromatin opening function that is an absolute requisite for transcription of this segment of the genome. In this manner phenotype is ultimately defined by gene potentiation.

MARs often flank the ends of genic domains encompassing various genic units. It is reasonable to propose that the remaining 30,000–35,000 pairs of the 100,000 MARs in each cell anchor the paired ends of the approx 12,000–30,000 genic domains to the nuclear matrix. It is likely that it is not a simple coincidence that this is also the number of genes transcribed in each cell. If MARs act, or participate as regions of locus control, then it is likely that their reiteration throughout the genome provides a means to specifically tag genes for potentiation in that cell.

The Human Genome Project (HGP) relies upon the availability of databases and tools that enable easy access, analysis, and comparison of genome information. Developing computational tools and algorithms that can assist in analysis, interpretation, and discovery of knowledge contained within these databases is critical. Specifically, these tools are expected to focus on completing the transcript map and understanding the functional significance of the sequenced genes. It is anticipated that elements of locus control, like MARs, will be sought during this phase given their key role in genetic processes and their localization to functional chromatin domains. Thus, a means to model these markers so that they could be placed on the genome sequence map would have significant ramifications.

Data mining and knowledge discovery techniques can be applied to genomic sequences to detect elements of locus control. MARs are one such type of locus-control element. Our limited knowledge of MARs has hampered formulating their detection using classical pattern recognition strategies where the existence of lower level constituent elements is used to establish the presence of a higher level functional block. However, MAR detection can be accomplished using the statistical estimation of "interestingness" of a sample. This detection strategy is of general utility for the detection of other classes of regulatory signals where a limited data set is available for describing the functional elements.

MARs have been experimentally defined for several gene loci, including, the chicken lysozyme gene, human interferon-β (IFN-β) gene, human β-globin gene, chicken α-globin gene, p53, and the human protamine gene cluster. Several motifs that characterize MARs have emerged, although a MAR consensus sequence is not apparent. The motifs that are currently utilized are functionally categorized and represented as AND-OR patterns described in the following.

The following characteristics of DNA known to be associated with the presence of Matrix Association Regions is summarized in Table 1.

- The Origin of Replication (ORI): It has been established that replication is associated with the nuclear matrix, and the origins of replication share the ATTA, ATTTA, and ATTTTA motifs.
- Curved DNA: Curved DNA has been identified at or near several matrix attachment sites and has been involved with DNA-protein interaction, such as recombination, replication and transcription. Optimal curvature is expected for sequences with repeats of the motif, *AAAAn$_7$AAAn$_7$AAAA* as well as the motif *TTTAAA*.
- Kinked DNA: Kinked DNA is typified by the presence of copies of the dinucleotide TG, CA, or TA that are separated by 2-4 or 9-12 nucleotides. For example, kinked DNA is recognized by the motif *TAn$_3$TGn$_3$CA*, with TA, TG and CA occurring in any order.
- Topoisomerase II sites: It has been shown that Topoisomerase II binding and cleavage sites are also present near the sites of nuclear attachment. Vertebrate and Drosophila topoisomerase II consensus sequence motifs can be used to identify regions of matrix attachment.
- AT-Rich Sequences: Typically many MARs contain stretches of regularly spaced AT-rich sequences in a periodic manner.
- TG-Rich Sequences: Some T-G rich spans are indicative of MARs. These regions are abundant in the 3' UTR of a number of genes and may act as recombination signals.
- Consensus Motif: The sequence TCTTTAATTTCTAATATATTTAGAA defined as the nuclear matrix STAB-1 binding motif.
- ATC Rule: ATC rule (a stretch of 20 or more occurrences of H, i.e., A or T or C). The ATC rule was used in the analysis of Rice A1-Sh2 region by some researchers. This rule has shown to be an effective indicator of regions with marked helix destabilization potential.
- Bipartite Signal: A bipartite sequence signature has been reported to be associated with the MARs. This is composed of two degenerate sequences AATAAYAA and AWWRTAANNWWGNNNC within a close proximity.
- A-Box, T-Box, etc., and other motifs that have been associated with MARs.

Their sequence interdependency is not known, but in most cases multiple motifs are utilized to create a functional MAR. Our current state of analysis makes it clear that at least three and possibly four independent types of MARs can be detected. These are MARs utilized as ORIs, (identified by the ORI rule), AT-rich MARs (identified by the AT-Rich rule and ATC rule) typified by lysozyme, or β-interferon genes. MARs that are not AT-Rich as exemplified by those of the protamine locus and the MARs that are tissue-specific.

Patterns and Rule Definitions

In the general framework discussed previously, a pattern description language must be defined that has sufficient power to represent the variety of patterns likely to be discovered as our understanding about DNA-protein interactions and the control of genetic machinery reaches a higher level of maturity. One must remember, each motif (and pattern) is represented by the probability of its random occurrence. This value can be derived using the base composition of the sequence being analyzed.

When searching for patterns, one must strive for a balance between the specificity and generality of patterns sought. Thus a distinction is often drawn between the tasks

Table 1
Table of Sequence Level Motifs[a]

Motif	*Motif name*	*DNA Signature*
m_1	ORI Signal	ATTA
m_2	ORI Signal	ATTTA
m_3	ORI Signal	ATTTTA
m_4	TG-Rich Signal	TGTTTTG
m_5	TG-Rich Signal	TGTTTTTTG
m_6	TG-Rich Signal	TTTTGGGG
m_7	Curved DNA Signal	AAAANNNNNNNAAAANNNNNNNAAAA
m_8	Curved DNA Signal	TTTTNNNNNNNTTTTNNNNNNNTTTT
m_9	Curved DNA Signal	TTTAAA
m_{10}	Curved DNA Signal	AAA
m_{11}	Curved DNA Signal	AAAA
m_{12}	Kinked DNA Signal	TANNNTGNNNCA
m_{13}	Kinked DNA Signal	TANNNCANNNTG
m_{14}	Kinked DNA Signal	TGNNNTANNNCA
m_{15}	Kinked DNA Signal	TGNNNCANNNTA
m_{16}	Kinked DNA Signal	CANNNTANNNTG
m_{17}	Kinked DNA Signal	CANNNTGNNNTA
m_{18}	mtopo-II Signal	RNYNNCNNGYNGKTNYNY
m_{19}	dtopo-II Signal	GTNWAYATTNATNNR
m_{20}	topo-II Vert. Signal	NCNNCYNGKTNYNY
m_{21}	AT-Cluster Signal	WWWWWW
m_{22}	AT-Cluster signal	AATATTTTT
m_{23}	AT-Cluster Signal	AATAAAYAAA
m_{24}	AT-Cluster Signal	ATATTT
m_{25}	ACBP-TF Signal	WTTTAYRTTTW
m_{26}	ARBP-TF Signal	ATTTCASTTGTAAAA
m_{27}	SAR-TF Signal	WWCAAWG
m_{28}	ARS 3'Consensus Signal	CTTTTAGCWWW
m_{29}	CEN Signal	TGTTTATGNTTTCCGAAANNNAAAA
m_{30}	HomProtcore Signal	TAATTA
m_{31}	YR-Richness Signal	YR
m_{32}	F/G Signal	AYCYRTRCAYYW
m_{33}	Bipartite AR Signal	AATAAYAA
m_{34}	Bipartite SAR Signal	AWWRTAANNWWGNNNC
m_{35}	MAMORI Signal	WAWTTDDWWWDHWGWHMAWTT
m_{36}	SAR Signal	WADAWAYAWW
m_{37}	SAR Signal	WWDAWAYAWW
m_{38}	SAR Signal	TWWTDTTWWW
m_{39}	T Box Signal	TTWTWTTWTT

[a] The set of motifs characterizing MARs constitute DNA-sequence signals or predicates upon which *rules* defining higher level patterns are constructed. Note that the IUPAC characters R, Y, W, and *k* are defined as: R = A or G, Y = T or C, *W* = A or T, and *k* = G or T.

of finding patterns and that of finding models or rules. The distinction between these two terms is rather arbitrary. Generally, a model is a global representation of a structure that summarizes components underlying the data that explain how the data may have arisen. In contrast, a pattern is a local structure, perhaps relating to a handful of variables and a few cases.

It is possible to employ a general set of DNA patterns using the AND-OR methodology. In such an AND-OR pattern specification methodology a disjunction (OR) of the conjunctions (AND) of the motifs detected in the sequence is used as the definition of the pattern being sought. The sequence level motifs serve as the lowest level predicates used to detect the presence of a higher level pattern. In general the following operations may be applied to the lower level motifs:

- Motif consensus sequence, m, represented as a regular expression of profile, or
- The logical OR of two motifs m_i and m_j , represented as $m_i \vee m_j$, or
- The augmented logical AND of two motifs m_i and m_j , represented as a $m_i \wedge_b^a m_j$ or
- The logical negation of a motif, m, represented as $\bar{m}$, specifying the absence of a given motif.

The pattern specification methodology must consider motif variability in its representation. Such variability may be captured using the AND-OR rules. As an example, consider the rule to define the Origin of Replication (OR) of DNA. This can be based on the OR operator applied to the three motifs m_1 =ATTA, m_2 = ATTTA, and m_3 = ATTTTA. The motif detectors bypass the AND layer in this case.

$$R_1 = m_1 \vee m_2 \vee m_3 \qquad \text{[Eq. 1]}$$

Similarly, the requirement for multiple motif occurrences can be specified using the AND operator. An additional parameter is incorporated when using the AND rule to constrain the allowable gap between the two co-occurring motifs. For example, the AT-Richness rule can be formulated as the occurrence of two hexanucleotide strings, m_4 = WWWWWW, that are separated by distance of 8–12 nt, using the augmented AND operator using $\wedge_{low}^{high}$ define the acceptable distance between the two motifs:

$$R_2 = m_4 \Lambda_8^{12} m_4 \qquad \text{[Eq. 2]}$$

The significance of the occurrence of a pattern in a DNA sequence is inversely related to the probability that the pattern will occur purely by chance. The probabilities of random occurrences of the underlying predicates are mathematically combined to evaluate the probability of the random occurrence of a pattern specified by a given rule. As an illustrative example, the random occurrence probabilities for the given patterns described by the above two rules can be computed. This value for the set of acceptable patterns described by rule R_2 is based on the occurrence of at least one motif within an acceptable distance from the reference motif. These probabilities are computed as shown in Tables 1 and 2.

In similar manner, the random occurrence probability rules are constructed on underlying predicates that are defined as profiles, can be computed using generating functions. As described, the rule probabilities are employed to estimate the statistical significance of matching the set of patterns that are detected in a given region of the DNA sequence.

Table 2
The Set of Biological Rules Defining Patterns That Were Used for Detecting Structural MARs.[a]

Rule	*Name*	*Definition*	*Probability*
R_1	ORI Rule	$m_1 \vee m_2 \vee m_3$	$p_1 = \Sigma_{i=1}^{3} \Pr(m_i)$
R_2	TG-Richness Rule	$m_4 \vee m_5 \vee m_6$	$p_2 = \Sigma_{i=4}^{6} \Pr(m_i)$
R_3	Curved DNA Rule	$m_7 \vee m_8 \vee m_9$	$p_3 = \Sigma_{i=7}^{9} \Pr(m_i)$
R_4	Kinked DNA Rule	$m_{12} \vee m_{13} \vee m_{14}$ $m_{15} \vee m_{16} \vee m_{17}$	$p_4 = \Sigma_{i=12}^{17} \Pr(m_i)$
R_5	Topoisomerase Rule	$m_{18} \vee m_{19} \vee m_{20}$	$p_5 = \Sigma_{i=18}^{20} \Pr(m_i)$
R_6	AT-Richness Rule	$m_{21} \wedge_{8}^{12} m_{21}$	$p_6 = \Pr(m_{21}) \cdot \{1 - \exp[-5 \cdot \Pr(m_{21})]\}$
R_7	MAR TF-Rule	$m_{25} \vee m_{26} \vee m_{27}$	$p_7 = \Sigma_{i=25}^{27} \Pr(m_i)$
R_8	AT Cluster Rule	$m_{21} \vee m_{22} \vee m_{23} \vee m_{24}$	$p_8 = \Sigma_{i=21}^{24} \Pr(m_i)$
R_9	Bipartite Rule	$m_{33} \wedge_{-200}^{+200} m_{34}$	$p_9 = \Pr(m_{33}) \cdot \{1 - \exp[-400 \cdot \Pr(m_{34})]\}$
R_{10}	SAR Rule	$m_{36} \vee m_{37} \vee m_{38}$	$p_{10} = \Sigma_{i=36}^{38} \Pr(m_i)$

[a] The table also specifies the relationship between the DNA-motif probabilities, $P(m_i)$, and the rule probabilities, p_j. These higher level statistical association forms the basis for mining MARs from DNA sequences.

Data Mining the MARs

There are two different data-mining methodologies. They are classified according to whether they seek to build models or find patterns. The methods aimed at building global models fall within the category of statistical exploratory analysis, which, for example, led to the rejection of conventional wisdom that long-term mortgage customers constitute a good portfolio. In a global sense, these customers constitute the group that were not able to find offers elsewhere and in fact may be the "not-so-good-customers." The second class of data-mining methods seeks to find patterns by sifting through the data seeking co-occurrences of specific values of specific variables. It is this class of strategies that has led to the notion of data-mining as seeking nuggets of information among the mass of data. The problem of detecting MARs from anonymous DNA sequence data falls within this category.

Data mining is the method employed to search for interesting patterns in data. Such a search often takes place in large data sets where the likelihood of finding such patterns is greater than expected. As described, the data-mining efforts aim at detecting statistically significant patterns, which are useful because they are not redundant, novel in regards to user's previous knowledge, simple for the user to understand, and sufficiently general to the referred population.

When searching for patterns, one must strive for a balance between the specificity and generality of the patterns sought. Thus, a distinction is often drawn between the tasks of finding *patterns* and that of finding *models*. The distinction between these two terms is rather arbitrary. Generally, a model is a global representation of a structure that summarizes components underlying the data that explain how the data may have arisen. In contrast, a pattern is a local structure, perhaps relating to a handful of variables and a few cases. Such local patterns are often sought in the time-series data analysis. One application of time series data analysis is the analysis of stock market data to detect interesting patterns that are novel, useful, and simple enough for the investors to understand.

There is a difference between the pattern-detection methods and the conventional diagnostic methods. One significant difference is that conventional diagnostic methods need a model to compare the data, while unsupervised pattern-detection does not require such a model. Another difference is the requirement in the pattern-detection context to search through very large collections of data and explore a large number of pattern shapes.

The search for MARs results from defining a group of patterns that are bonded together in order to form a biologically functional unit that is classified by similar function. After grouping, a search for the patterns from a given group in the query DNA sequence can be performed. If a large subset of members of a functionally related group of patterns is found in a specific region of the uncharacterized DNA sequence, one can begin to learn about its function. This process is called a *Functional Pattern Search* and is typified by the MAR-Wiz system that performs a search for the group of patterns that are associated with Matrix Attachment Regions (MARS).

It is quite intuitive to consider pattern-cluster density as a property defined along the span of a sequence. A *sliding window* algorithm can be applied for measuring this value, where the measurements are characterized by the two parameters, W and δ. The cluster-density is measured in a window of size W centered at location x along that sequence. Successive window measurements are carried out by sliding this window in the increments of δ nucleotides. If δ is small, linear interpolation can be used to join the individual window estimates that are gathered at x, $x + \delta$, ..., $x + k\delta$. In this manner, a continuous distribution of the cluster-density is obtained as a function of x.

The task of estimating the density of pattern clusters in each window can be statistically defined as a functional inverse of the probability of rejecting the null hypothesis, that states that the frequency of the patterns observed in a given window is not significantly different from the expected frequencies from a random W nucleotide sequence of the same composition as the sequence being analyzed. The inverse function chosen as, $\rho = -\log(\alpha)$, where the parameter α is the probability of erroneously rejecting H_0. In other words, α represents the probability that the set of patterns observed in a window occurred purely by chance. The value of ρ is computed for both the forward and the reverse DNA strands because we do not know which strand or if both strands will be bound, the average of the two values is considered to be the density estimate at a given location.

In order to compute ρ, assume that we are searching for k distinct types of patterns within a given window of the sequence. In general, these patterns are defined as rules $R_1, R_2, ..., R_k$.The probability of random occurrence of the various k patterns is calculated using the AND-OR relationships between the individual motifs. Assume that these probabilities for k patterns are $p_1, p_2, ..., p_k$. Next, a random vector of pattern frequencies, F, is constructed. F is a k-dimensional vector with components, $F = \{x_1, x_2, ..., x_k\}$, where each component x_i is a random variable representing the frequency of the pattern R_i in the W base-pair window. The component random variables x_i are assumed to be independently distributed Poisson processes, each with the parameter $\lambda_i = p_i \cdot W$. Thus, the joint probability of observing a frequency vector $F_{obs} = \{f_1, f_2, ..., f_k\}$ purely by chance is given by:

$$\mathrm{P}(F_{obs}) = \prod_{i=1}^{k} \frac{e^{-\lambda_i}\lambda^{f_i}}{f_i!} \quad \text{where} \quad \lambda_i = p_i \cdot W \qquad \text{[Eq. 3]}$$

The steps required for computation of α, the cumulative probability that pattern frequencies equal to or greater than the vector F_{obs} occurs purely by chance is given by Equation 4 below. This corresponds to the one-sided integral of the multivariate Poisson distribution and represents the probability that the H_0 is erroneously rejected.

$$\begin{aligned}\alpha &= P_r\,(x_1 \geq f_1, x_2 \geq f_2, \ldots, x_k \geq f_k) \\ &= P_r\,(x_1 \geq f_1) \cdot P_r\,(x_2 \geq f_2) \;\ldots.\; P_r\,(x_k \geq f_k) \\ &= \sum_{x_1 = f_1}^{\infty} \frac{\exp^{-\lambda_1}\lambda_1^{x_1}}{x_1!} \cdot \sum_{x_2 = f_2}^{\infty} \frac{\exp^{-\lambda_2}\lambda_2^{x_2}}{x_2!} \\ &\quad \cdots\cdots\; \sum_{x_K = f_K}^{\infty} \frac{\exp^{-\lambda_K}\lambda_k^{x_K}}{x_k!}\end{aligned} \qquad \text{[Eq. 4]}$$

The p-value, α, in Equation 4 is utilized to compute the value of ρ or the cluster-density as specified in Equation 5 below:

$$\begin{aligned}\rho = \ln \frac{1.0}{\alpha} = -\ln(\alpha) = \sum_{i=1}^{k} \lambda_i + \sum_{i=1}^{k} \ln f_i\,! - \sum_{i=1}^{k} f_i \ln \lambda_i \\ - \sum_{i=1}^{k} \ln\left(1 + \frac{\lambda_i}{f_i + 1} + \frac{\lambda_i^2}{(f_i+1)(f_i+2)} + \ldots. + \frac{\lambda_i^t}{(f_i+1)(f_i+2)\,\ldots..(f_i+t)}\right)\end{aligned} \qquad \text{[Eq. 5]}$$

The infinite summation term in Equation 5 quickly converges and thus can be adaptively calculated to the precision desired. For small values of λ_i, the series may be truncated such that the last term is smaller than an arbitrarily small constant, ε.

Using MAR-Wiz

Figure 1, presents the output from the analysis of the human β-globin gene sequence. In Fig. 1A Rules 1 through 6, or the core rules, were utilized for the detection of MARs in this sequence. All Rules 1 through 10 were utilzed for detection of MARs shown in Fig. 1B. This statistical inference algorithm based on the association of patterns found within the close proximity of a DNA sequence region has been incorporated in the MAR-Wiz tool. A java-enabled version of the tool is available for public access (*see* Website: http://www.futuresoft.org/MARWiz).

We observe that the results in Fig. 1A and B are in agreement. Fig. 1B appears noisy. A possible explaination for this is an interdependence between the some of the new rules, i.e., Rules 7 through 10, on the core rules. Thus, when a core rule is detected in the sequence, an associated new rule is also present. However, the MAR-potential assumed this event to be significant as independence of MAR rules is assumed. Consequently, the MAR-potential and noise increase. The interdependence of the MAR rules needs to be clarified.

MAR-Wiz was originally conceived to examine sequences from cosmid sized (~ 40 kb) sequencing projects, the initial backbone of human genome project. The program default values of window length (1000), step size (100), and run length (3), i.e., number of concurrent steps were implemented accordingly. For cosmid sized or larger sequences, step size should be maintained at 100 bp, to minimize noise while permitting sufficient discrimination.

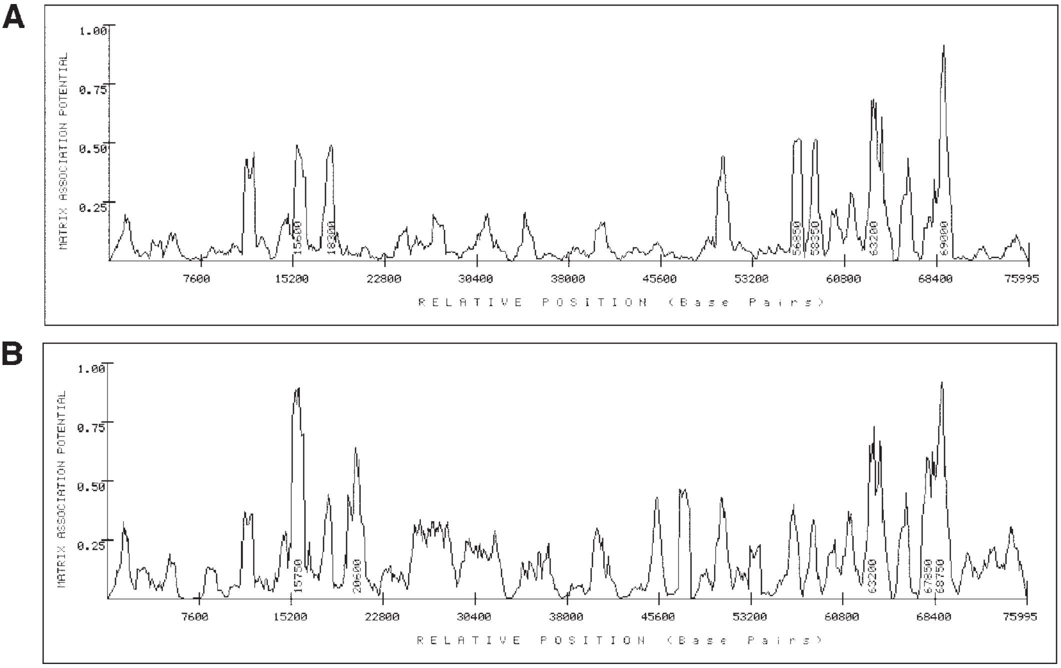


Fig. 1. The analysis of human beta-globin gene cluster using the MAR-Wiz tool. Default analysis parameters were used for generating the plot in **(A)**. Similar results are obtained when Rules R_7 through R_{10} were also included. These are shown in **(B)**. The results with all rules included tended to be more noisy, thereby indicating that the occurrence of the new rules was dependent on the old rules.

In the case of shorter sequences ranging in size from 1,000–10,000 bp, a window size of 100 with a step of 10 usually yields optimum results. This is best illustrated below in the analysis of the IFN-β gene shown in Fig. 2. The predicted locations of regions of matrix attachment correspond well with those experimentally determined and the predictions by SIDD (Stress Induced DNA Destabilization) program.

Several generalizations regarding the utility and interpretation of the data analysis by MAR-Wiz have become apparent over the course its development and use as an analytical tool. As discussed earlier, one of the significant contributions from these analyses has been the classification of candidate MARs as a function of the distribution of MAR sequence motifs they possess. This is best exemplified by AT-rich MARs and ORI MARs. Both are examples of MARs with different biological functions acting as either boundary elements or as origins of replication respectively. This has led to the strategy of initially scanning the query sequence with all available rules as a means to establish the potential class of MARs contained within the sequence selected for analysis. The distribution of identified elements and their local concentration can then be used to guide their initial classification and subsequent assessment of biological function. Experience has shown that the initial classification scan is most efficiently carried out in three phases. First, sequences are examined with all the core (default setting) rules selected to establish the overall distribution of motifs. This is subsequently followed by scanning with the ORI and AT-rich rules deselected in the presence and absence of the ATC rule. The relative contribution of each rule to the

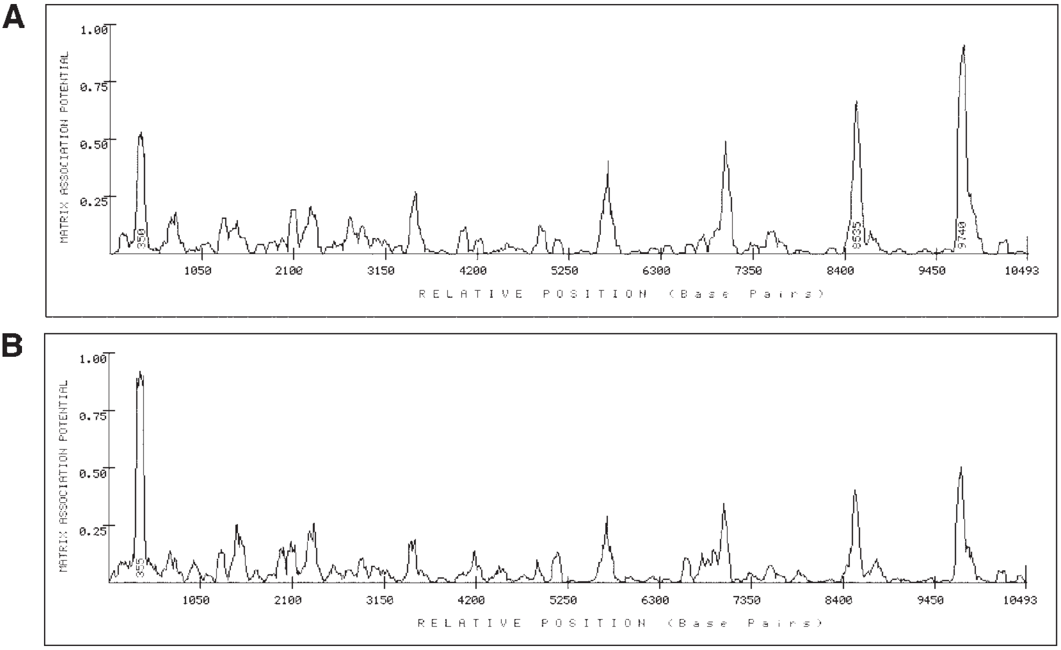


Fig. 2. For shorter sequences, such as the Human-interferon gene shown above, the window size of 100 and step size of 10 is optimal. In this case as well, the augmented Rule set in **(A)** does not indicate any additional regions with higher MAR potential. An increase in the strength of the first peak in **(B)** in fact relatively lowers the significance of the remaining peaks.

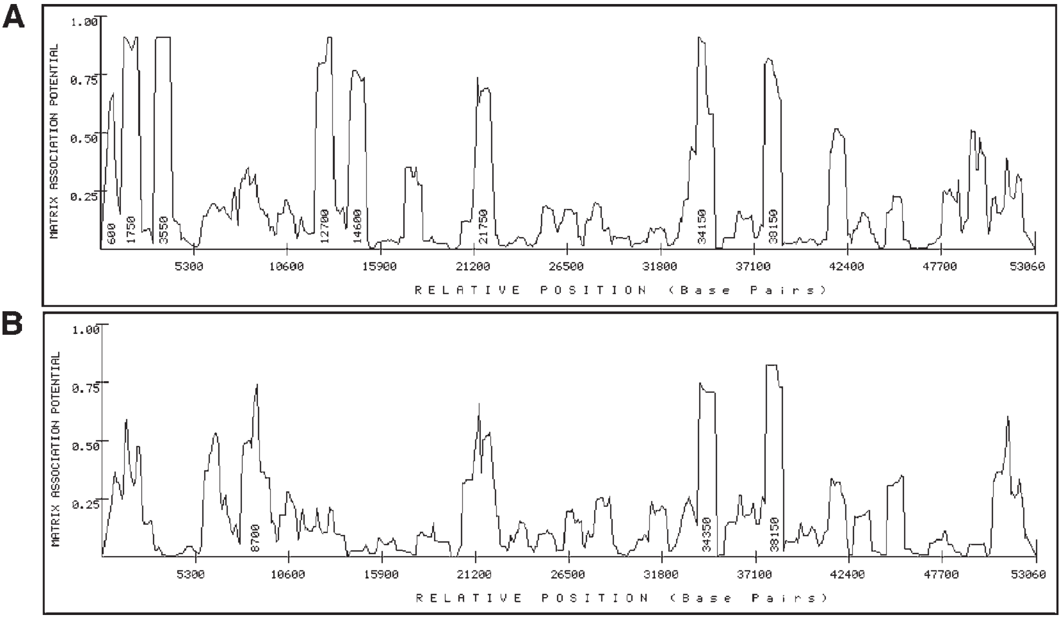


Fig. 3. The results of analyzing the human protamine gene cluster using the MAR-Wiz algorithm. **(A)** shows the potential with all core rules selected. **(B)** shows the analysis results with all default core rules selected, minus the ORI and AT-Richness rule.

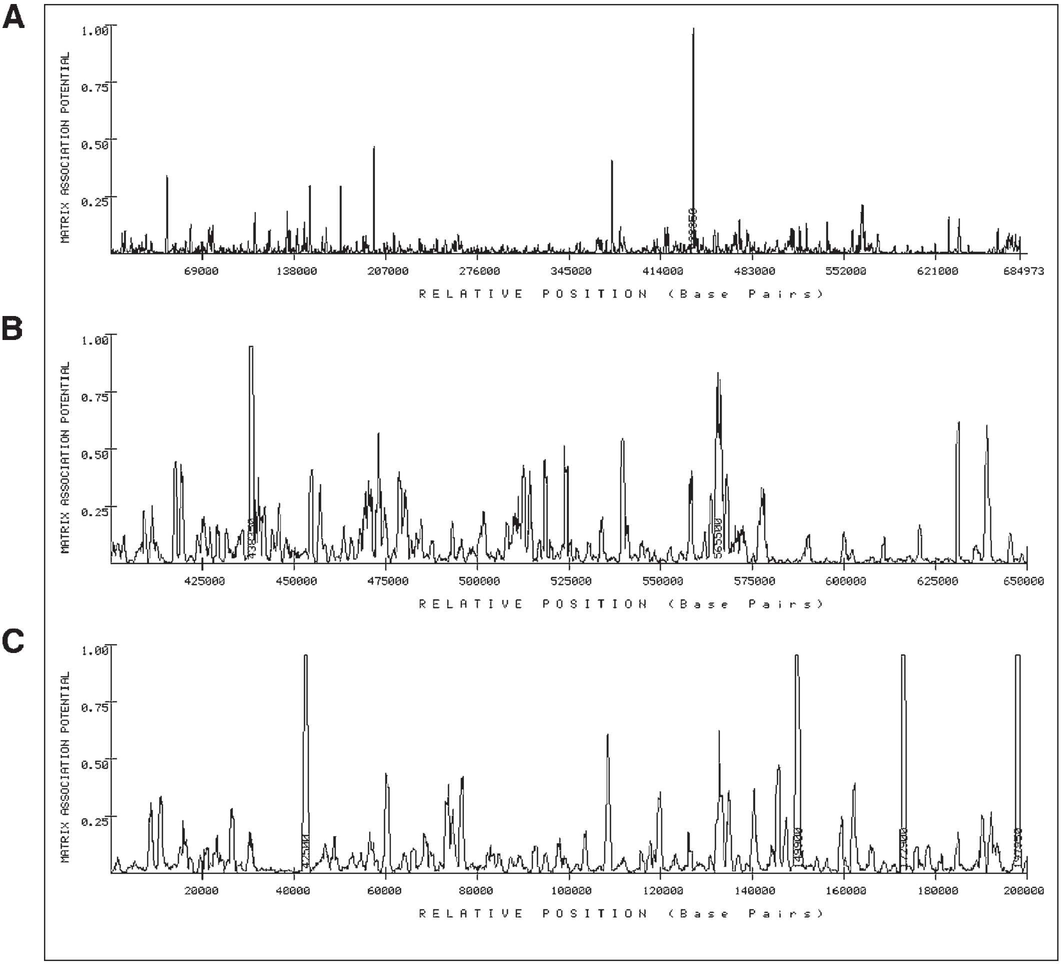


Fig. 4. The results of analyzing the human T-cell receptor gene with MAR-Wiz. In **(A)**, the large peak (with the absolute potential value of 120) overshadows the lower, albeit significant, peaks. This peak is clipped to a value of 30 and the other peaks normalized accordingly. This enables us to visualize the locations of other candidate MARs in the two ends of this locus, as shown in **(B)** and **(C)**.

detection of that motif within the query sequence can then be assessed. Having, used the rule parameter selection to initially categorize the MAR, one then examines the distribution of probabilities using a low-threshold of detection. While this strategy may yield a high false-positive rate, it ensures the detection of most MARs even when the individual MAR is at the limits of statistical detection (possess an expected value is at least e^{-8}).

As shown in Fig. 3, this strategy was effectively utilized to identify and classify the MARs of the PRM1 → PRM2 → TNP2 locus to the sperm-specific class. It must be emphasized that the predictions of MAR-Wiz can only provide a guide to biological verification.

As shown in Fig. 4A, a clustering of motifs with a low likelihood of occurrence by chance within any given region results in a single small region of very high MAR

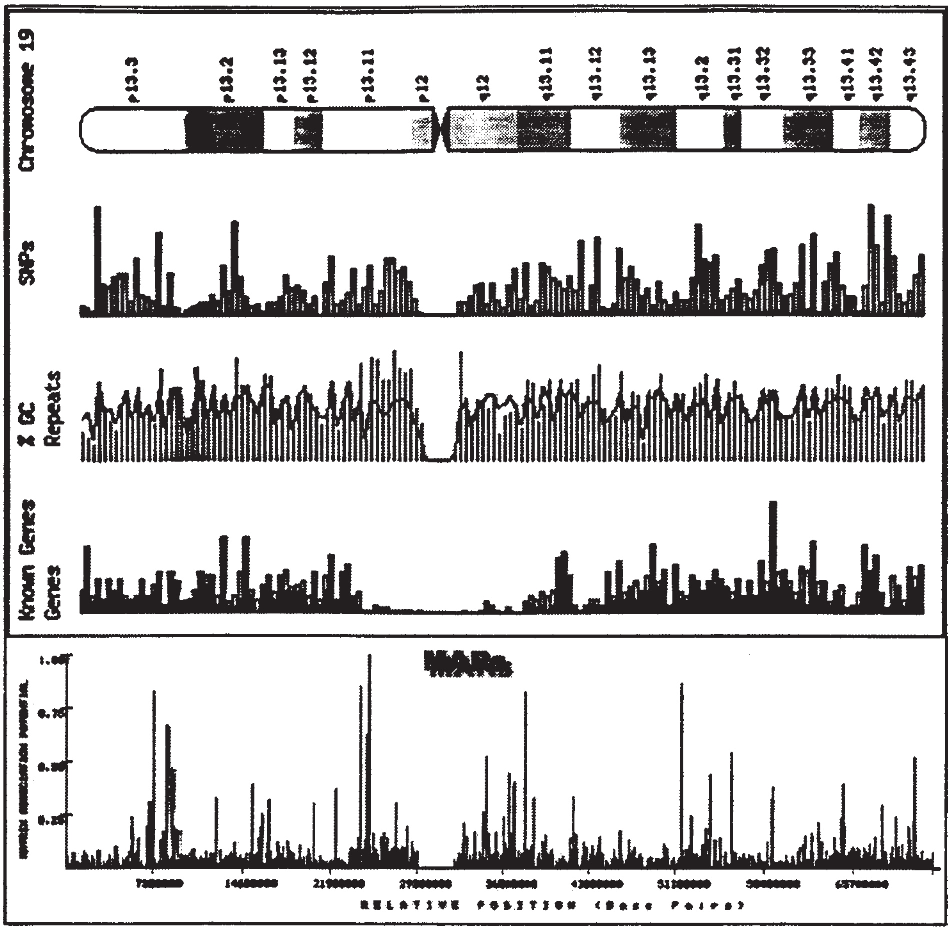
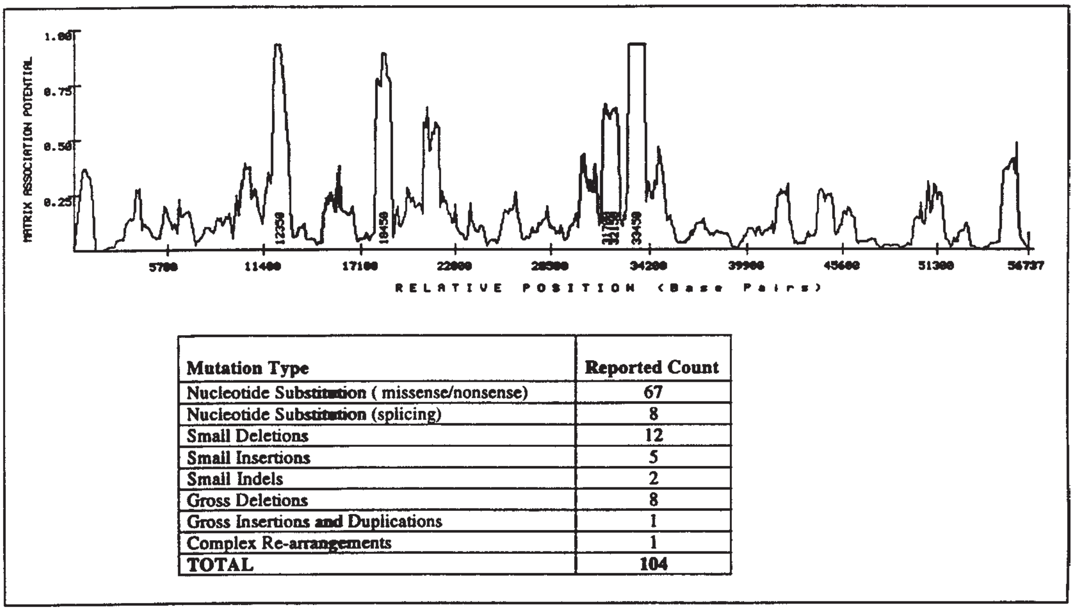


Fig. 5. Analysis of chromosome 19 using the MAR-Wiz software. The results have been aligned with the SNP, GC, Repeat, and data about the known genes available at ENSEMBL (*see* Website: www.ensembl.org). The examples in the chapter present applications of MAR-Wiz to classes of sequences where excellent correspondence when compared to the results obtained by wet-bench analysis was observed. However, there are cases where improvement is required.

potential. As is the case with the T-cell receptor gene locus, the one peak visible has an absolute potential value of 120. This corresponds to a chance occurrence of the observed motifs of approx 10^{-40}. During the normalization process, the other segments containing statistically significant candidate MARs are visually suppressed and upon initial inspection appear insignificant. This MAR-Wiz tool can effectively compensate for this artifact by adjusting the saturation value of the display, i.e., the peak height to which all values are clipped and result the saturation (i.e., 100%) of data values. The data is then scaled to 100% of this value and visualization of the previously masked segments is then apparent. This correction was performed for the two ends of the T-Cell receptor gene and is shown in Figs. 4B and Fig. 4C. In both of these cases, the potential values were clipped to 30, i.e., all values of MAR-Potential higher than 30 were saturated to this value. The remaining values were then normalized using this potential. The location of potential MARs has thus become apparent.



Mutation Type	Reported Count
Nucleotide Substitution (missense/nonsense)	67
Nucleotide Substitution (splicing)	8
Small Deletions	12
Small Insertions	5
Small Indels	2
Gross Deletions	8
Gross Insertions and Duplications	1
Complex Re-arrangements	1
TOTAL	104

Fig. 6. The analysis of the Lesch-Nyhan syndrome HPRT gene shows that additional research is required to identify MARs such as those embedded in introns (positions 5534-6107). Interestingly, most peaks are coincident with known mutations in this human gene (*see* Website: http://www.uwcm.ac.uk/uwcm/mg/search/119317.html).

Figure 5 presents the MAR-Wiz results of analyzing the chromosome 19. These results have been overlaid with the existing information available on chromosome 19 at ENSMBL (*see* Chapter 25; Website: http://www.ensembl.org).

To illustrate how Mar-Wiz can be applied, we will consider an example problem and its solution. Consider *hprt* gene known to be responsible for Lesch-Nyhan syndrome, a devastating self-mutilating disease. Analyze this gene sequence using MAR-Wiz at Website: http:///www.futuresoft.org/MARWiz. Correlate the existence of MARs with the known human mutations causing this disease (*see* Fig. 6).

In this case a MAR has been biologically defined to be contained within the first intron (positions 5534-6107), and has shown great promise when incorporated as an integral component of a gene therapy. However, this MAR region was not identified by the software. This may reflect the fact that this MAR functions as an ARS (Autonomously Replicating Sequence), which is yet another type of MAR.

Irrespective, other candidate MAR sequences have been identified by MAR-Wiz. It is interesting to note that most of these candidates are coincident with known human mutations causing this disease. This would suggest that their biological assessment is warranted.

Glossary and Abbreviations

CDNA Complementary DNA. Reverse transcriptase may be used to synthesize DNA that is complementary to RNA (e.g., an isolated mRNA).

Chromatin Stainable material of interphase nucleus consisting of nucleic acid and associated histone protein packed into nucleosomes.

Circular DNA DNA arranged as a closed circle. This causes serious topological problems for replication that are resolved by DNA topoisomerase.

DNA Deoxyribonucleic acid. The genetic material of all cells and many viruses. A polymer of nucleotides. The monomer consists of phosphorylated 2-deoxyribose N-glycosidically linked to one of four bases : adenine, cytosine, guanine or thymine. The sequence of bases encodes genetic information.

DNA Binding Proteins Proteins that interact with DNA, typically to pack or modify the DNA (e.g., Histones), or to regulate gene expression (e.g., transcription factors).

DNA Replication The process whereby a copy of a DNA molecule is made, and thus the genetic information it contains is duplicated.

DNA Topoisomerase An enzyme capable of altering the degree of supercoiling of double-stranded DNA molecules. Various topoisomerases increase or relax supercoiling, convert single-stranded rings to double-stranded rings, tie and untie knots, catenate and decatenate rings.

DNA Transfection This term is a hybrid of transformation and infection and generally denotes the introduction of other kinds of genes or gene fragments into cells as DNA. For example, introduction of activated oncogenes from tumours into tissue culture cells.

DNAse Deoxyribonuclease. An endonuclease with preference for DNA. In chromatin, the sensitivity of DNA to digestion by DNAase I depends on its state of organization, transcriptionally active genes being much more sensitive than inactive genes.

Enhancer A DNA control element frequently found 5' to the start site of a gene, which when bound by a specific transcription factor, enhances the levels of expression of the gene, but is not sufficient alone to cause expression.

Euchromatin The chromosomal regions that are diffuse during interphase and condensed at the time of nuclear division.

Facultative Heterochromatin That heterochromatin which is condensed in some cells and not in others, possibly representing differences in the activity of genes in different cells.

Gene Originally defined as the physical unit of heredity. Best defined as the unit of inheritance that occupies a specific locus on a chromosome. Given the occurrence of split genes, it might be re-defined as the set of DNA sequences (exons) that are required to produce a single polypeptide.

Gene Expression The full use of the information encoded in a gene via transcription and translation leading to production of a protein leading to the appearance of the phenotype determined by that gene. Gene expression is controlled at various points in this process. This control is thought to be the major determinant of cellular differentiation in eukaryotes.

Gene Locus The position of a gene in a linkage map or on a chromosome.

Gene Therapy Treatment of a disease caused by malfunction of a gene, by stable transfection of the cells of the organism with the normal gene.

Heterochromatin The chromosomal regions that are condensed during interphase and at the time of nuclear division.

Junk DNA Genomic DNA that serves, as yet, no known function.

Nuclear Envelope A membrane surrounding the nucleus of eukaryotic cells. Consists of inner and outer membranes separated by perinuclear space and perforated by nuclear pores.

Nuclear Lamina A fibrous protein-network lining the inner surface of the nuclear envelope. Proteins of the lamina are lamins A, B and C, which have sequence homology to proteins of intermediate filaments.

Nuclear Matrix Protein latticework filling the nucleus that anchors required for DNA replication and transcription complexes.

Promoter A region on the DNA that RNA polymerase binds to before initiating the transcription of DNA into mRNA. The nucleotide at which transcription starts is designated +1. Nucleotides positions with negative numbers indicate upstream nucleotides and with positive numbers indicate downstream nucleotides. Most factors that regulate gene transcription do so by binding at or near the promoter and affecting the initiation of transcription. RNA polymerase II, that transcribes all genes that code for polypeptides, recognizes many thousands of promoters. Most have the Goldberg-Hogness or TATA box centered around position –25 and has the consensus sequence 5'-TATAAAA-3'. Several promoters have a CAAT box around position –90 with the consensus sequence 5'-GGCCAATCT-3'. There is increasing evidence that all promoters for genes for housekeeping proteins contain multiple copies of a GC-rich element that includes the sequence 5'-GGGCGG-3'. Transcription by polymerase II is also affected by more distant elements known as enhancers.

Repetitive DNA Nucleotide sequences in DNA present in the genome in numerous copies. These sequences are not thought to code for polypeptides. One class of repetitive DNA, termed highly repetitive DNA, is found as short sequences (5–100 nucleotides in length) repeated thousands of times in a single long stretch. It typically comprises 3–10% of the genomic DNA and predominantly constitutes the satellite DNA. Another class, which comprises 25–40% of the DNA and termed moderately repetitive DNA, usually consists of sequences about 150–300 nucleotides in length that are dispersed evenly throughout the genome. This includes the Alu sequence and transposons.

Transcription mRNA synthesis mediated by RNA-Polymerases utilizing the DNA as a template.

Transcription Factor Protein required for recognition by RNA polymerases of specific stimulatory sequences in eukaryotic genes. Several are known that activate transcription by RNA polymerase II when bound to upstream promoters.

Suggested Readings

MARs: Background and Data Models

Benham, C., Kohwi-Shigematsu, T., and Bode, J. (1997) Stress-induced duplex DNA destabilization in scafold/matrix attachment regions, J. Mol. Biol. 274, 181–196.

Kramer, J. and Krawetz, S. (1997) PCR-Based assay to determine nuclear matrix association, Biotechniques 22, 826–828.

Kramer, J., Adams, M., Singh, G., Doggett, N., and Krawetz, S. (1998) Extended analysis of the region encompassing the PRM1→PRM2→TNP2 domain: genomic organization, evolution and gene identification, J. Exp. Zool. 282, 245–253.

Singh, G., Kramer, J., and Krawetz, S. (1997) Mathematical model to predict regions of chromatin attachment to the nuclear matrix, Nucleic Acid Res. 25, 1419–1425.

Singh, G., Stamper, D., and Krawetz, S. (1998) A web tool for detecting matrix association regions, Trends Genet. 14(2), 8.
Java version is available *(see* Website: http://www.futuresoft.org/MAR-Wiz).

van Drunen, C., Sewalt, R. G., Oosterling, R. W., Weisbeek, P. J., Smeekens, S. C., and van Driel, R. (1999) A bipartite sequence element associated with matrix/scaffold attachment regions, Nucleic Acids Res. 27, 2924–2930.

van Drunen, C., Oosterling, R. W., Keultjes, G. M., Weisbeek, P. J., van Driel, R., and Smeekens, S. C. (1997) Analysis of the chromatin domain organization around the plastocyanin gene reveals a MAR-specific sequence element in Arabidopsis thaliana, Nucleic Acids Res. 25, 3904−3911.

Staden, R. (1988) Methods for calculating the probabilities of finding patterns in sequences, Comput. Appl. Biosci. 5(2), 89–96.

The Biological Description of Matrix Attachment Regions

Bode, J., Stenger-lber, M., Kay, V., Schalke, T., and Dietz-Pfeilstetter, A. (1996) Scaffold/matrix attachment regions: Topological switches with multiple regulatory functions, Crit. Rev. Eukaryot. Gene Expr. 6(2,3), 115−138.

Deppert, W. (1996) Binding of MAR-DNA elements by mutant p53: Possible implications for oncogenic function, J. Cell. Biochem. 62, 172−180.

Farache, G., Razin, S., Targa, F., and Scherrer, K. (1990) Organization of the 3'-boundary of the chicken alpha globin gene domain and characterization of a CR 1-specific protein binding site, Nucleic Acid Res. 18, 401−409.

Jade, J., Rios-Ramirez, M., Mielke, C., Stengert, M., Kay, V., and Klehr-Wirth, D. (1995) Scaffold/Matrix attachment regions: structural properties creating transcriptionally active loci, Intl. Rev. Cytol. 162A, 389−454.

Jarman, A. and Higgs, D. (1988) Nuclear scaffold attachment sites in the human globin gene complexes, EMBO J. 7(11), 3337−3344.

Kramer, J. and Krawetz, S. (1996) Nuclear matrix interactions within the sperm genome, J. Biol. Chem. 271(20), 11619−11622.

Phi-Van, L. and Strätling, W. (1988) The matrix attachment regions of the chicken lysozyme gene co-map with the boundaries of chromatin domain, EMBO J. 7, 655−664.

Von Kries, J., Phi-Van, L., Diekmann, S., and Strätling, W. (1990) A non-curved chicken lysozyme 5' matrix attachment site is 3' followed by a strongly curved DNA sequence, Nucleic Acid Res. 18, 3881−3885.

MAR Sequence Patterns

Boulikas, T. (1993) Nature of DNA sequences at the attachment regions of genes to the nuclear matrix, J. Cell. Biochem. 52, 14–22.

Hartwell, L. and Kasten, M. (1994) Cell cycle control and cancer, Science 266, 1821–1828.

Sander, M. and Hsieh, T. (1985) Drosophila topoisomerase II double stranded DNA cleavage: analysis of DNA sequence homology at the cleavage site, Nucleic Acid Res. 13, 1057–1067.

Spitzner, J. and Muller, M. (1988) A consensus sequence for cleavage by vertebrate DNA topoisomerase II, Nucleic Acid Res. 16(12), 5533–5556.

Strissel, P., Espinosa, III, R., Rowley, J., and Swift, H. (1996) Scaffold attachment regions in centromere-associated dna, Chromosoma 105, 122–133.

Data Mining

Crecone, N. and Tsuchiya, M. (1993) Special issue on learning and discovery in databases, IEEE Trans. Knowl. Data Eng. 5(6).

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996) From data-mining to knowledge discovery: An overview, in: Advances in Knowledge Discovery and Data Mining, (Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthuruswamy, R., eds.), AAAI Press, Menlo Park, CA, pp. 1–34.

Hand, D. (1998) Data mining: Statistics and more? The American Statistician 52(2), 112–118.

Klosgen, W. (1992) Problems for knowledge discovery in databases and their treatment in the statistics interpreter EXPLORA, Intl. J. Intell. Sys. 7(7), 649–673.

Klosgen, W. (1995) Efficient discovery of interesting statements in databases, J. Intell. Info. Sys. 4(1), 53–69.

Higher Order Control of Transcription

Kadonaga, J. (1998) Eukaryotic transcription: An interlaced network of transcription factors and chromatin-modifying machines, Cell 92, 307–313.

Kliensmith, L. and Kish, V. (1995) Principles of cell and molecular biology, 2nd ed, HarperCollins, Williamsport, PA.

Krawetz, S., Kramer, J., and McCarry, J. (1999) Reprogramming the male gamete genome: a window to successful gene therapy, Gene, 234, 1–9.

Roeder, R. (1998) The role of general initiation factors in transcription by RNA polymerase II, Trends Biochem. Sci. 21, 327–335.

24 Analyzing Sequences Using the Staden Package and EMBOSS

Rodger Staden, David P. Judge, and James K. Bonfield

Introduction

Since the beginning of big genome sequencing, initiated by the work on the nematode *Caenhorhabditis elegans*, the Staden group has concentrated on developing methods to increase the efficiency of these large-scale projects. In the course of this, we have designed and implemented a sophisticated and intuitive graphical user interface for use in our programs GAP4 and PREGAP4. This interface has also been used in our sequence analysis program SPIN, but as it has not been the main focus of our efforts, SPIN is still limited in the number and variety of the functions it contains. The EMBOSS project was initiated to provide a comprehensive set of sequence analysis tools that would be available free to all and has made rapid progress towards this goal. However, it did not have a graphical user interface and this limited its usefulness. It was felt that the combination of SPIN and EMBOSS would provide a powerful package.

To make it possible to have a single uniform graphical user interface to the programs in the EMBOSS package, we first combined our two sequence analysis programs NIP4 and SIP4. One dealt with analyzing individual nucleic acid sequences and the other comparing pairs of nucleic acid or protein sequences. Next we developed ways to launch EMBOSS programs and handle their output. Two features in the design of EMBOSS facilitated our work: 1) its flexibility in output formats and 2) its use of a language (ACD) for specifying the inputs to its programs. The first technical challenge was to parse the ACD to automatically produce suitable dialogue boxes for each EMBOSS program and to prepare SPIN to load the results into memory. The second problem was to parse these varying results files to display the results and allow users to interact with them as though they had been produced by internal SPIN functions.

SPIN is available free to academic sites for UNIX and for Microsoft operating systems (*see* Website: http://www.mrc-lmb.cam.ac.uk/pubseq/). EMBOSS is available for UNIX operating systems (*see* Website: http://www.hgmp.mrc.ac.uk/Software/EMBOSS/).

Introduction to SPIN

SPIN is an interactive and graphical program for analyzing nucleotide sequences. It contains functions to search for restriction sites, consensus sequences/motifs and protein coding regions. In addition, it can analyze the composition of the sequence and perform translations. It also has a good set of sequence comparison functions for both nucleic acid and proteins. Most functions produce both textual and graphical results.

Introduction to EMBOSS

EMBOSS is a free Open Source software analysis package specially developed for the needs of the molecular biology-user community. The software uses data in a variety of formats and even allows transparent retrieval of sequence data from the web. As extensive libraries are provided with the package, it is a platform that allows other scientists to develop and release software in true open source spirit. EMBOSS also integrates a range of currently available packages and tools for sequence analysis into a seamless whole. At the time of writing EMBOSS contained over 100 programs for: 1) sequence alignment; 2) rapid database searching with sequence patterns; 3) protein motif identification (including domain analysis); 4) nucleotide sequence pattern analysis, (for example to identify CpG islands or repeats); 5) codon usage analysis for small genomes; 6) rapid identification of sequence patterns in large scale sequence sets; and 7) presentation tools for publication.

Setting up EMBOSS for SPIN

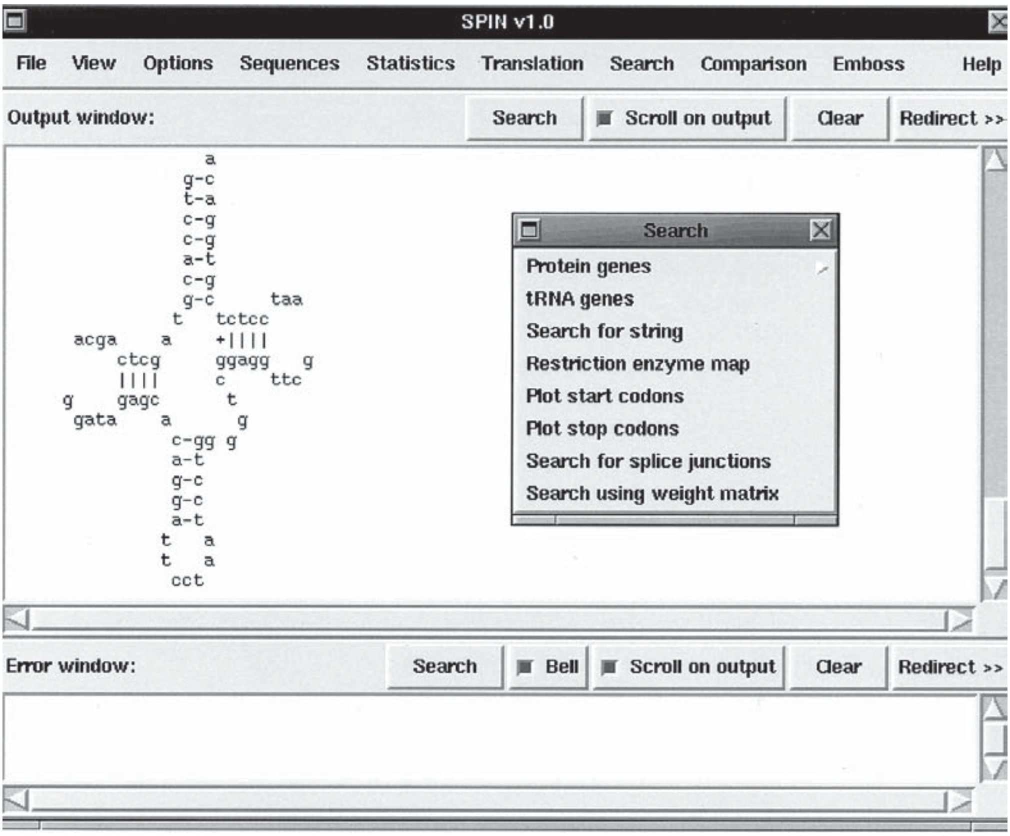
Both SPIN and EMBOSS can be downloaded from their respective ftp sites and installed as independent packages. In order to use SPIN as an interface to EMBOSS, it is necessary to create dialogues for all EMBOSS programs. This is performed by the single command **create_emboss_files**, which is included with SPIN. The EMBOSS menu will then appear in SPIN and its programs will be available.

SPIN User Interface

Spin has a graphical user interface with four main displays. The first is a window from which all the main options are selected and which receives textual results. Most analytical functions add their graphical results either to a *Sequence Plot* window that is associated with the sequence being analysed, or for cases where two sequences are being compared, to a *Sequence Comparison Plot* window. (An exception is the restriction enzyme search that produces its own separate window.) Each of these graphical windows has an associated *Sequence Display* window for viewing the sequences. The main window contains an *Output* Window for textual results, an *Error* window for error messages and a series of menus arranged along the top. Figure 1 shows an example of the main window in which the *Output* Window contains a result from the tRNA gene search. The contents of the two text windows can be searched, edited and saved. Each set of results is preceded by a header containing the time and date when it was generated.

SPIN Sequence Plot

Figure 2 shows several graphical results displayed in a *Sequence Plot* overlaid with a *Sequence Display* window and a pop-up menu. The top four panels show results from protein gene prediction methods and below that are matches from a motif search. The first coding prediction method produces results for each of the three reading frames, which include short vertical lines showing the positions of

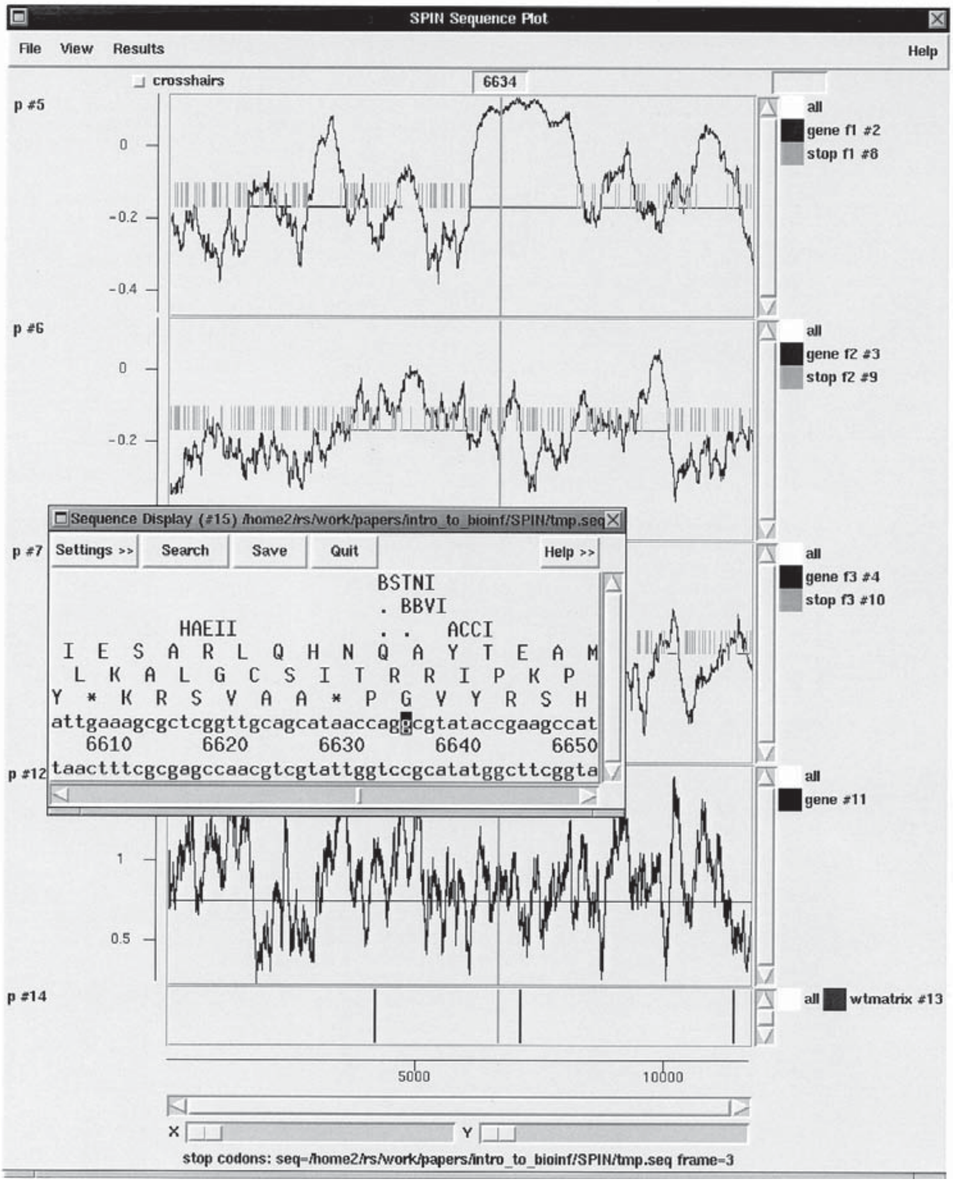


```
             a
           g-c
           t-a
           c-g
           c-g
           a-t
           c-g
           g-c     taa
          t   tctcc
 acga    a    +||||
     ctcg     ggagg   g
     ||||     c    ttc
g    gagc      t
 gata    a      g
          c-gg g
          a-t
          g-c
          g-c
          a-t
         t   a
         t   a
          cct
```

Fig. 1.The SPIN main window including the results from a tRNA gene search.

stop codons in each frame and a continuous line representing the likelihood of coding based on codon usage. Below is a coding prediction result that only has a single value for each position along the sequence. The bottom panel contains short vertical lines showing the positions and scores for a motif search. Each *Sequence Plot* has a cross-hair and scrollbars for zooming and moving the results in the x and y directions. The position of the cross-hair is shown in the boxes above the plot. The x position is shown in sequence base numbers in the left-hand box above the plot, and the y coordinate, expressed using the score values of the gene search, is shown in the right hand box. At the right hand side of each panel is a set of square boxes with the same colors as the lines drawn in the adjacent plot. These icon-like objects represent individual results and allow the user to operate on them via pop-up menus. As a minimum, these menus contain the commands: **Information**, **List results**, **Configure**, **Hide** and **Remove**, but as will be seen later, some sets of results have additional options. These icons can also be used to drag and drop the results to a new location.

As shown in Fig. 2, each Sequence Plot window also has an associated Sequence Display window which can be invoked by double clicking on the plot. The Sequence Plot contains a cursor that denotes the position of the cursor in the Sequence Display. The user can move the cursor and this will move the cursor in the Sequence Display and all other displayed cursors that relate to the sequence.



```
                            BSTNI
                            . BBVI
           HAEII            . .   ACCI
 I  E  S  A  R  L  Q  H  N  Q  A  Y  T  E  A  M
  L  K  A  L  G  C  S  I  T  R  R  I  P  K  P
Y  *  K  R  S  V  A  A  *  P  G  V  Y  R  S  H
attgaaagcgctcggttgcagcataaccaggcgtataccgaagccat
   6610      6620      6630      6640      6650
taactttcgcgagccaacgtcgtattggtccgcatatggcttcggta
```

stop codons: seq=/home2/rs/work/papers/intro_to_bioinf/SPIN/tmp.seq frame=3

Fig. 2. The SPIN Sequence Plot including two protein gene search results, a motif search and a Sequence Display.

SPIN Sequence Display

From the Sequence Display (*see* Fig. 2), the user can view the sequence in textual form. For DNA, one or both strands can be shown as well as the encoded amino acids. Restriction enzyme sites can also be shown and simple text string searches performed to scroll the sequence. The position of the sequence display cursor is shown in the Sequence Plot, from where it can be driven using the mouse.

SPIN Sequence Comparison Plot

The *Sequence Comparison Plot* shows the results of comparison algorithms, including local and global alignments. Each match is represented as either a single dot or a line, depending on the analysis performed. As for the *Sequence Plot*, sets of matches from a single invocation of a comparison command are termed a *result*. Each result is plotted using a single color, which can be configured via the *Results Manager* (*see* Subheading "Spin Results Manager").

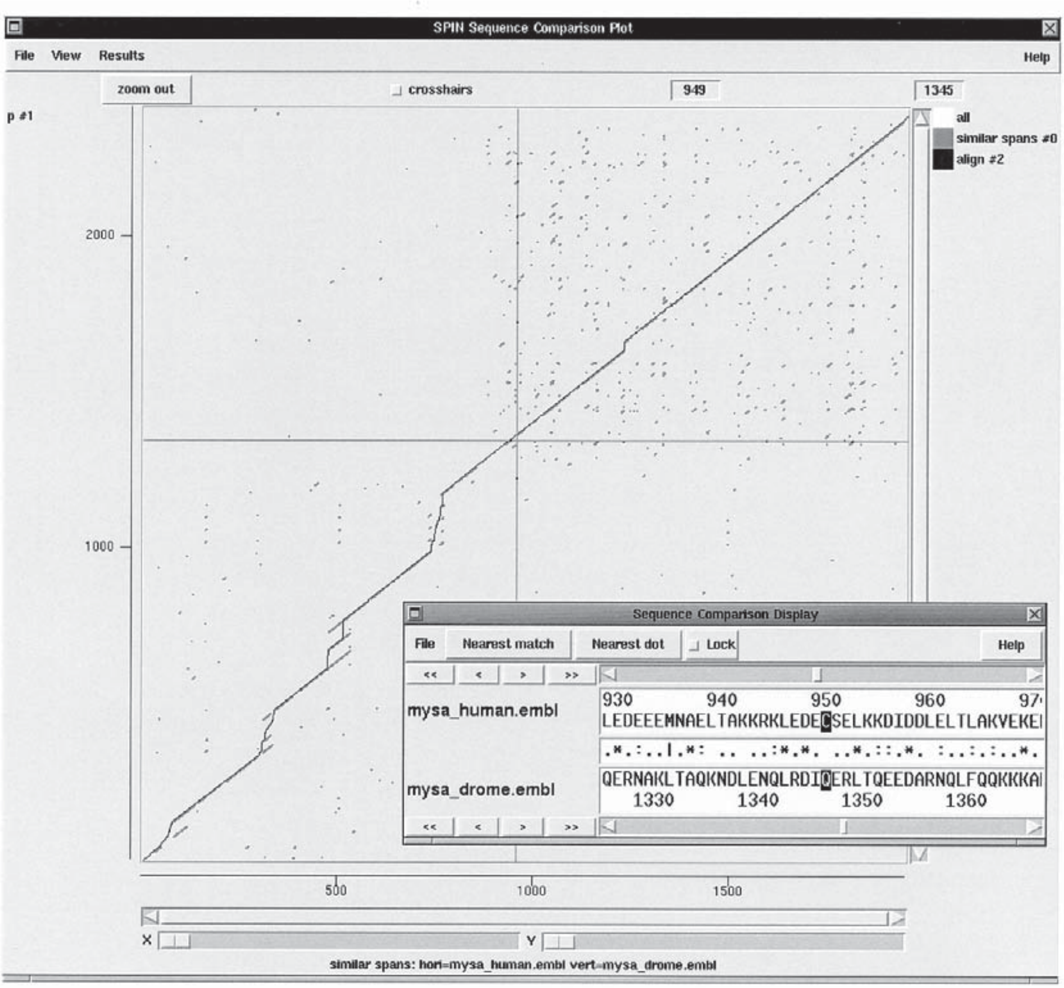
Vertical and horizontal rulers are plotted around the edge of the display and cross-hairs can be used to find the location of any position. It is possible within SPIN to compare many different sequences so there may be more than one horizontal or vertical sequence shown in the *Sequence Comparison Plot*. All the points are scaled to the largest sequence in each direction. As for the *Sequence Plot*, results can be dragged and dropped between plots. Each set of results has an associated *Sequence Comparison Display*, the cursors of which can be seen and controlled from within the *Sequence Comparison Plot*. An example *Sequence Comparison Plot* plus an overlaid *Sequence Comparison Display* is shown in Fig. 3. This contains the result of applying the **Find matching spans** algorithm and the **Align Sequences Globally** function to EMBL sequence library entries mysa_drome and mysa_human. This first algorithm finds and plots (as dots) similar segments of the two sequences and the second finds the best alignment between them and plots its path (as line segments). The results show that they are closely related. The figure also includes cross-hairs and the cursors from the *Sequence Comparison Display*.

SPIN Sequence Comparison Display

Figure 3 shows the Sequence Comparison Display superimposed on the Sequence Comparison Plot. The two sequences are shown one above the other with similarities marked in the intervening strip. The sequences can either be scrolled independently, or in register (by selecting the **Lock** button). As the display is associated with a particular set of results, the **Nearest match** and **Nearest dot** buttons can be used to jump the cursor to neighboring matches. This can make it much easier to examine the detail of individual aligned segments in the Sequence Comparison Plot. *Nearest match* means the match whose x,y coordinate in sequence character positions is closest, whereas *Nearest dot* means the match that appears closest in screen coordinates. (If the scaling is the same in both directions the *Nearest dot* and *Nearest match* will be equivalent.) The display can also be scrolled by dragging the cursor in the *Sequence Comparison Plot*.

SPIN Restriction Enzyme Map

The *SPIN* restriction enzyme map function finds and displays restriction sites found within a specified region of a sequence. As shown in Fig. 4, the results are plotted in a Restriction Enzyme Map. This window has some different capabilities from the other SPIN displays. Each record in the plot corresponds to a particular enzyme: their names are written to the left and their cut sites marked opposite as short vertical lines. The results can be scrolled vertically (and horizontally if the plot is zoomed in). A ruler is shown along the base and the current cursor (the vertical black line) position is shown in the left-hand box near the top right of the display. If the user clicks, in turn, on two restriction sites, their separation in base pairs will appear in the top right-hand box. Information about the last site touched is shown in the information line at the bottom



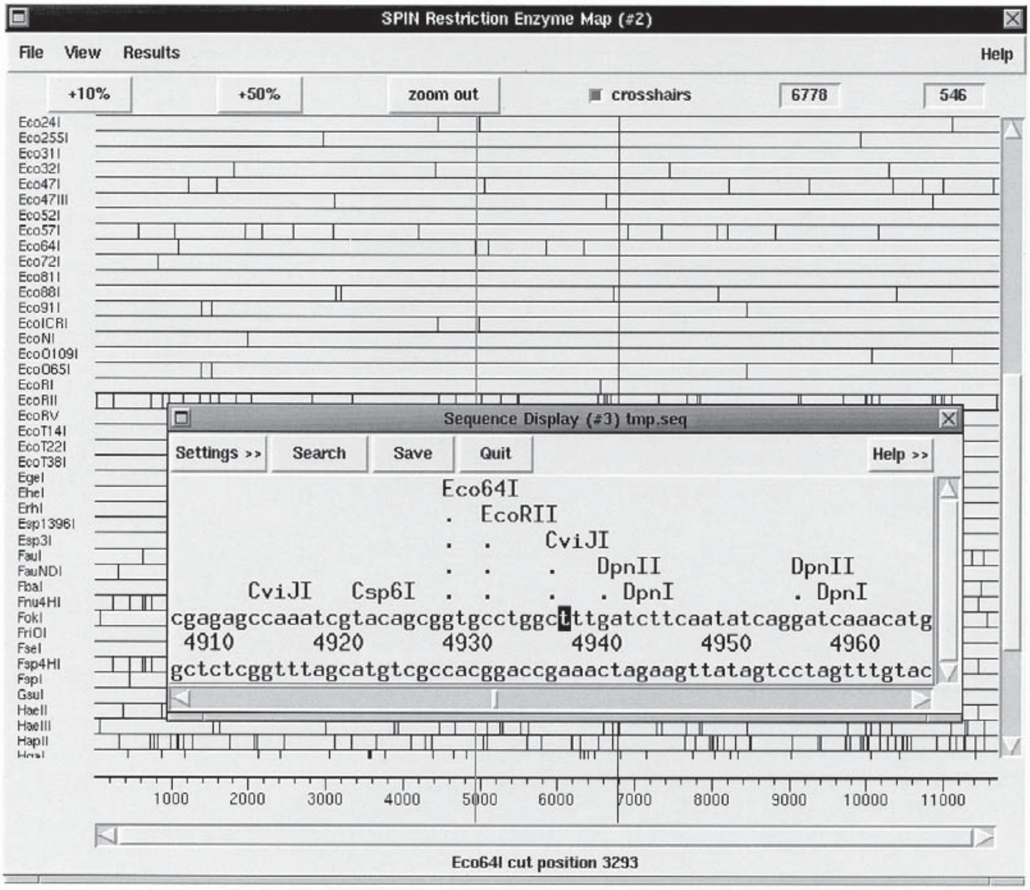
```
                  930       940       950       960       97
mysa_human.embl   LEDEEEMNAELTAKKRKLEDECSELKKDIDDLELTLAKVEKE
                  .*.:..|.*: .. ..:*.*. ..*.::.*. :..:.:..*.
mysa_drome.embl   QERNAKLTAQKNDLENQLRDIQERLTQEEDARNQLFQQKKKA
                     1330      1340      1350      1360
```

Fig. 3. The SPIN Sequence Comparison Plot including results for similar segments, an alignment path, and a Sequence Comparison Display.

of the display. The order of the records can be changed using drag and drop. As for the Sequence Plot, the Restriction Enzyme Map has an associated Sequence Display whose cursor can be seen and controlled in its display. From the View menu of the Restriction Enzyme map, the results can be written to the SPIN Output window.

SPIN Sequence Manager

SPIN manages sequences at two levels. First it provides for access to read sequences into the program, and second, it contains a range of facilities for deriving new sequences from them. For example, it can produce the complement of a DNA sequence, rotate it about any position, translate it, or scramble it. Each of these types of internal operations produces a new sequence that can be analyzed using the SPIN functions, or that can be saved to disk. SPIN is limited to reading sequences in plain text, EMBL, GenBank, PIR,



```
                        Eco64I
                        .  EcoRII
                        .  .    CviJI
                        .  .    .   DpnII          DpnII
      CviJI    Csp6I    .  .    .   . DpnI         . DpnI
cgagagccaaatcgtacagcggtgcctggcttgatcttcaatatcaggatcaaacatg
 4910      4920      4930      4940      4950      4960
gctctcggtttagcatgtcgccacggaccgaaactagaagttatagtcctagtttgtac
```

Fig. 4. The SPIN Restriction Enzyme Map including a Sequence Display.

and FASTA formats. Through its interface to EMBOSS, it can also read sequences from remote sequence libraries. Once loaded, the sequences are selected and manipulated using the *Sequence Manager*, which is available from the SPIN Sequences menu.

SPIN Results Manager

Most SPIN functions produce *results*. The *Result Manager*, which can be accessed from the main menu or the individual plots, provides a mechanism to interrogate and manipulate these results. Each result can be listed showing the time it was created, the name of the function that created the result, and a unique identifier. A pop-up menu can be used to obtain further information, list the results in the Output window, configure its plot (change its color, line width), hide or reveal its plot, or remove the result altogether.

SPIN Analytical Functions

SPIN's analytical functions are accessed through the following menus: 1) Statistics, 2) Translation, 3) Search, and 4) Comparison. The EMBOSS functions have their own separate menu.

SPIN Statistics Menu

This menu contains simple functions to write the base and dinucleotide composition of nucleic acid sequences to the Output Window and to plot their base composition to a Sequence Plot.

SPIN Translation Menu

This menu contains functions to set the genetic code, translate to protein sequences, to find open reading frames and to calculate and write codon tables to disk files.

SPIN Set Genetic Code Function

This function allows the user to change the genetic code used in all the options. The codes are defined in a set of codon table files distributed with the package, but calculated from those maintained by the National Center for Biological Information, USA. The user simply selects the list of code names from a dialogue box. The selection can be made for the current run of SPIN or made a permanent setting for future runs.

SPIN Translation Function

In addition to the translation that can be shown in the Sequence Display, SPIN can write a translation to the Output Window. This translation can either be for all six frames, or using an EMBL style feature table, limited to defined segments. The contents of the feature tables are displayed in a dialog box and the user can select which CDS records to translate. The translation is written to the Output Window in FASTA format, from where it can be saved to a file.

SPIN Find Open Reading Frames Function

Using the current genetic code, this function will find open reading frames greater than a specified length. As shown in Fig. 5 the results can be output in two ways, either in feature table format or FASTA format. The user can select the start and end points to do the search, which strand to search (either the forward, reverse, or both) and the minimum length of the open reading frame in codons. If the output is being written in the FASTA format, the name of a file is also required.

SPIN Codon Usage Function

Codon usage tables can be calculated and written to the Output window and written to disk. If required, the values found can be added to the counts in a preexisting codon table, or when written out to disk they can be concatenated with an existing codon table file. In the first case the existing file will be read and added to the values calculated for the region defined by the user. In the latter, the values calculated for the region defined by the user will be written immediately after those from the existing table, hence producing a pair of tables joined end to end. The protein gene searches: **Codon Usage Method** and **Author Search** use single or double codon tables. The values in the table can be expressed as observed counts or as percentages of usage for the cognate amino acid.

SPIN Search Menu

The SPIN Search menu contains simple functions like the restriction enzyme, stop codon, start codon, and subsequence searches, plus some more complex functions for finding genes and motifs. It should be noted that these gene-finding methods, although

```
FT                  CDS 512..736
FT                  CDS 525..965
FT                  CDS 740..952

>512                512..736
CLTLSLKESFIRHAAYLEGSRSEKRDVCVARESKRCSEASARSVTGGDSKWIAVQPQRPL
LGRLCNKRGPGSLSA*
>525                525..965
ALKKVLYDTRHTSKGAGVKNVMSVSLVSRNVARKLLLVQLLVVIASGLLFSLKDPFWGVS
AISGGLAVFLPNVLFMIFAWRHQAHTPAKGRVAWTFAFGEAFKVLAMLVLLVVALAVLKA
VFLPLIVTWVLVLVVQILAPAVINNKG*
>740                740..952
RFVYDICLASPGAYTSERPGGLDIRIWRSFQSSGDVGVTGGGVGGFKGGILAADRYVGFG
AGGSDTGTGCN*
```

Fig. 5. The SPIN Open Reading Frame Results in Feature Table and FASTA formats.

pioneering when invented, have since received little attention, and more recent methods may be more effective and better tested.

SPIN Restriction Enzyme Search

Files of restriction enzyme names, recognition sequences, and their cut sites are stored in disk files supplied with the package, but users can edit them to produce their own. Three default files, plus a browser for loading user files are made available from the function's dialogue box. Once loaded, the enzyme names are presented in a scrollable dialogue. The results are presented as shown in Fig. 4. A similar dialogue is used to configure the restriction enzymes shown in the Sequence Display. The results can be shown in textual form in the Output Window, from where they can be saved to disk.

SPIN Subsequence Search

Two subsequence or string searches are available in SPIN. The first is selected from the **Search** menu on the Output window, producing both graphical and textual output. The second is selected from the **Search** button in the Sequence display, it moves the cursor to the position of the next match. For DNA sequences, the user can define a subsequence using the IUPAC codes, set the percentage match required and which strand to search. The search can be made literal but is never case-sensitive. The results are plotted in a Sequence Plot and written to the Output Window.

SPIN Stop and Start Codon Search

A stop codons search can be carried out on either (or both) strands of the sequence. The stop codons are displayed graphically, with a different color used for each reading frame, and their positions can also be listed in the Output window. If any of the three phase gene search methods (described in the following) are currently being displayed, the stop codons will automatically be plotted on top of the corresponding frame. A similar function locates and plots the positions of start codons. These searches use the current set of genetic code tables.

SPIN Motif Search

Through the IUPAC symbols, the subsequence search described earlier allowed the use of special characters to encode permitted sets of sequence characters at each position in the search string. The SPIN Motif Search is more flexible in that it allows different

```
Mount acceptors
     18    15   0.0   10.0
 P -14 -13 -12 -11 -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3
 N 113 113 113 113 113 113 113 113 113 113 113 113 113 113 113 113 113 113
 T  58  50  57  59  67  56  58  49  47  66  64  31  34   0   0  11  41  31
 C  21  28  34  25  29  33  35  32  42  40  33  25  74   0   0  23  28  41
 A  17  11  11  18   7  17  12  23  15   3  10  29   5 113   0  24  21  21
 G  17  24  11  11  10   7   8   9   9   4   6  28   0   0 113  55  23  20
```

Fig. 6. A SPIN DNA Weight Matrix

scores to be assigned to each character type at each position along a subsequence or motif. These scores are stored in the form of *weight matrices*. These must be created beforehand from sets of aligned sequences that are known to contain the motif. This is usually done using the program *make_weights*, which is supplied with the package.

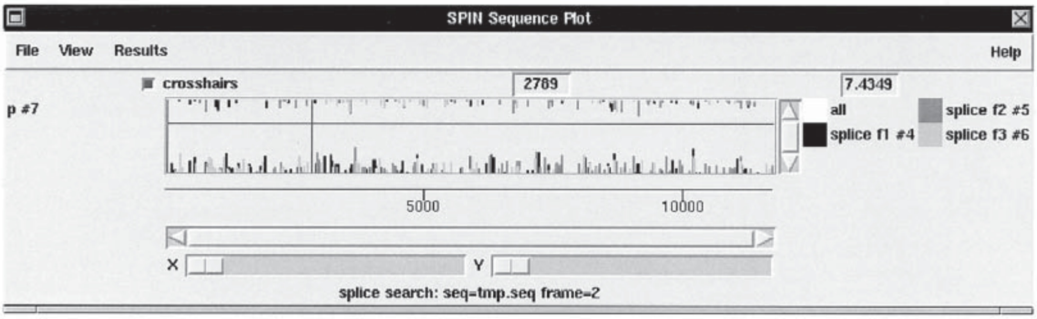
An example weight matrix file is shown in Fig. 6. It consists of a title record—a record defining the motif size, an offset and the score range—two records that need to be present but that are ignored; and four records (for DNA) defining the base counts calculated from trusted examples. These counts are converted into weights that are used during the searches. Any position in a sequence that scores at least as high as the minimum score is reported as a match, and if the results are plotted they are scaled to fit the range defined by the minimum and maximum scores. A typical plot is shown at the base of Fig. 2.

SPIN Splice Site Search

The SPIN splice junction search uses a pair of weight matrices to search for splice junctions. It differs from the Motif Search described above only in the way the results are plotted. The results are displayed in three colors, one color for each reading frame. The donors are plotted upwards from the base of the panel and the acceptors are plotted downwards from the top of the panel. The donors and acceptors with the same color are compatible; e.g., the same colored donors are compatible with the same colored acceptors. Of course it is the combination of reading frame and splice sites that really matters, so donors and acceptors drawn in different colors can be compatible if the reading frame changes. By default all the sites are drawn in the same plot (*see* Fig. 7). They can be separated by reading frame using the program's ability to reorganize the graphical results. This layout of the donors and acceptors is designed to add to the protein gene search methods and stop codon plots. The results are plotted as log-odds.

SPIN Codon Usage Gene Search

Although the codon usage gene search method contains some improvements over those of the original publication, this gene finding method is essentially that previously described by Staden (1984). For all protein gene-finding methods we are trying to decide if each segment of the sequence is coding or noncoding. Usually, each possibility is represented by a table of expected codon usage and the calculation finds the odds that each segment of the sequence fits either the coding or noncoding table and the results are presented as log odds in a Sequence Plot. At each position along the sequence the program also plots a single dot for the reading frame with the highest score. These dots appear at the midpoints of the three panels and will form a continuous line if one reading frame is consistently the highest scoring. An example is shown in Fig. 2.



splice search: seq=tmp.seq frame=2

Fig. 7. The SPIN Splice Junction Search Plot.

The user supplies the name of a file containing two concatenated codon usage tables, the first being from coding sequence and the second from noncoding sequence. This double codon table can be calculated by SPIN using the Codon Usage function described earlier. If the user gives the name of a file that contains only a single codon table the algorithm will assume that it is from coding sequence and will generate a noncoding table that consists of the frequencies that would be expected if the sequence being analyzed was random but had the same base composition as the supplied codon table.

If no table is specified the program will generate a codon usage table corresponding to an average amino acid composition and then derive a noncoding table from its base composition. In addition the user can select to set the amino acid composition of the coding table to have an average amino acid composition, and/or to have no codon preference (i.e., for each amino acid the codon counts are equal, e.g., for the standard genetic code [TTT = TTC]; [TTA = TTG = CTT = CTC = CTA = CTG];...; [GGT = GGC = GGA = GGG]). In the latter case the search uses amino acid composition only.

SPIN Author Test Gene Search

This Codon Usage method uses a different mathematical treatment based on methods used to decide authorship of text, i.e., is the usage of words (codons) more like that of author A (coding) or that of author B (noncoding)? Again the main input is a pair of concatenated codon tables and the results are plotted in the same manner. However, the algorithm calculates the optimal weighting to give each codon to obtain the best discrimination between coding and noncoding sequence. The user sets the expected error rate as a percentage and the algorithm will choose the corresponding window length to use for the analysis.

SPIN Base Bias Gene Search

This method for finding protein genes, unlike the methods already described, does not attempt to say either which strand or frame is likely to be coding, only which regions of the sequence. The method analyzes the frequencies of each of the four base types in each of the three positions in long stretches of codons. Ficket (1982) showed that the bias in these measures can be used to indicate possible protein coding regions. The level of bias is plotted on a scale that shows the probability that the sequence is coding. An example is shown in the bottom panel in Fig. 2.

```
score   9 probability 1.73e-04 expected  365 observed 1772
score  10 probability 1.17e-05 expected   25 observed  601
score  11 probability 3.60e-07 expected    1 observed  149
```

Fig. 8. A SPIN Tabulate Scores Result

SPIN tRNA Gene Search

This method is used to find segments of a sequence that might code for tRNAs. It looks for potential cloverleaf forming structures and then for the presence of the expected conserved bases. It presents results in a graphic Sequence Plot in an identical manner to those for a motif search, and draws out the cloverleaf structures in the Output Window (*see* Fig. 1).

SPIN Comparison Menu

SPIN contains three functions for finding local segments of similarity between pairs of sequences (Find Similar Spans, Find Matching Words, and Local Alignment), plus a global alignment algorithm. It can also compare a nucleic acid sequence to a protein by automatically calculating a three-frame translation and can compare DNA against DNA at the protein level. In both cases the results are superimposed using a different color for each phase, hence avoiding the possibility of reading frame shifts obscuring any similarity. All of these functions produce results that are plotted in the Sequence Comparison Plot described earlier. Alignments and matching segments can be examined at the sequence level using the Sequence Comparison Display. The score matrices and other associated values can be configured in the SPIN Options menu. The combination of algorithms and graphical user interface make this one of the stronger features of SPIN. A further strength is that the program contains algorithms for calculating the probabilities of observing the hits found by two of its comparison functions.

Comparison Function Probabilities

For the *Find Similar Spans* and *Find Matching Words* functions, the program enables users to compare observed and expected hits to help assess the significance of their results. These probability calculations are also used to set the default scores for these methods. The probability depends on the composition of the two sequences, the cut-off score and for the Find Similar Spans algorithm, the score matrix. The probability calculated is the chance of finding the given score in infinitely long random sequences of the same composition as those that are being compared. The expected number of matches for any score is calculated by multiplying its probability value by the product of the lengths of the two sequences. The matches found for these two algorithms can be assessed by selecting the **Tabulate Scores** option from the pop-up menu in the Sequence Comparison Plot, which will write a list of observed and expected results in the Output Window (*see* Fig. 8). As shown in Fig. 8, there are clearly many more matches at each score level than would be expected by chance.

Find Similar Spans

This method calculates a score for each position in the plot by summing points found when looking forwards and backwards along a diagonal line of a given (window) length. The algorithm uses a score matrix that contains scores for every possible pair of character types. At each point that the score is above a minimum score, a match is plotted.

If one of the sequences is DNA and the other protein, the program will automatically calculate a 3-frame translation of the DNA and plot the results for each translation in a different color. The dialogue allows the user to set the window length and cut-off score. From the result icon pop-up menu available in the Sequence Comparison Plot the user can request: *Information*, *List results*, *Tabulate scores*, and *Rescan matches*; in addition to the standard *Display sequences*, *Configure*, *Hide*, and *Remove*. *Information* gives a brief description of the sequences used, the input parameters and the number of matches found. *List results* writes every segment of alignment in the Output Window. *Rescan* matches revisits each matching span and plots a dot for each residue pair whose score is above a given threshold. An example is shown in Fig. 3.

Find Matching Words

The *Find Matching Words* function finds regions of identical characters shared by the two sequences. Its main value is speed, being very much faster than the Find Similar Spans function. It is not very sensitive but is useful for long DNA sequences. Users set only the minimum word length. The pop-up from the Sequence Comparison Plot is the same as that for Find Matching Spans.

Align Sequences Locally

The SPIN local alignment routine is based on the program SIM by Huang and Miller. SIM finds the best nonintersecting alignments between two sequences or within a single sequence using dynamic programming. The algorithm requires space proportional to the sum of the input sequence lengths and the output alignment lengths. The user can either specify the number of alignments to find or that all alignments above a given score should be reported. Gap open and gap extend values are also required. The alignments are displayed in the Output Window and also on the Sequence Comparison Plot as a series of lines showing its path.

Align Sequences Globally

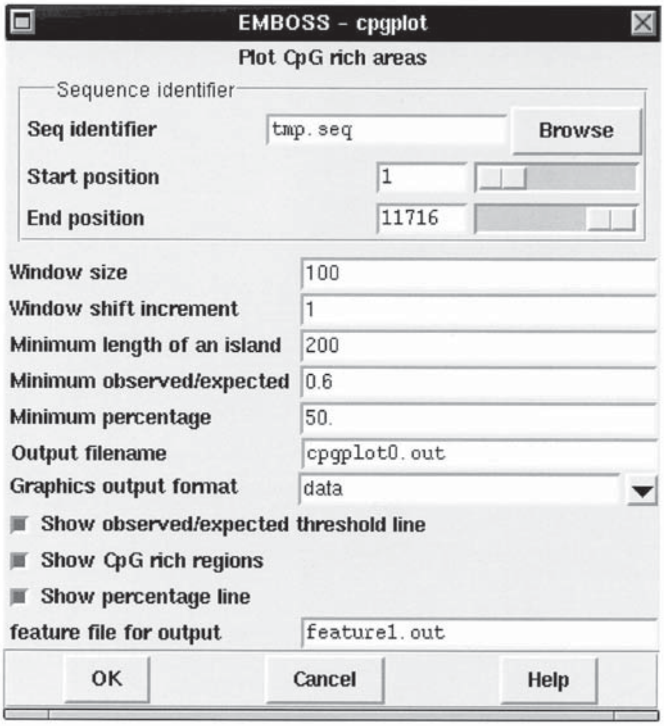
The SPIN global dynamic programming alignment algorithm is based on an algorithm of Huang, which is a linear space method. Users can supply gap open and gap extend penalties. The alignment is written to the Output Window and its path drawn in the Sequence Comparison Plot. An example is shown in Fig. 3.

EMBOSS Functions

EMBOSS contains a comprehensive set of programs for sequence analysis. In this section we give two illustrations of the SPIN interface and a shortened list of the EMBOSS applications. Using the EMBOSS programs through SPIN provides a uniform graphical interface through which applications can be selected, configured via appropriate dialogue boxes, and from which the results can be viewed and manipulated intuitively. The EMBOSS applications are selected from the SPIN EMBOSS menu.

An Example of SPIN EMBOSS Dialogue

Figure 9 shows the SPIN dialogue for the EMBOSS application **cpgplot**. It includes a browser for using the Sequence Manager (described earlier) to select the sequence to analyze, and boxes for setting its start and end points. Below are boxes for setting various parameters and output choices. The **Graphics output format** is selected using a pop-up menu by clicking on the arrow. The default value *data* will



EMBOSS - cpgplot

Plot CpG rich areas

Sequence identifier

Seq identifier: tmp.seq

Start position: 1

End position: 11716

Window size: 100

Window shift increment: 1

Minimum length of an island: 200

Minimum observed/expected: 0.6

Minimum percentage: 50.

Output filename: cpgplot0.out

Graphics output format: data

■ Show observed/expected threshold line

■ Show CpG rich regions

■ Show percentage line

feature file for output: feature1.out

Fig. 9.The SPIN dialogue for the EMBOSS cpgplot program.

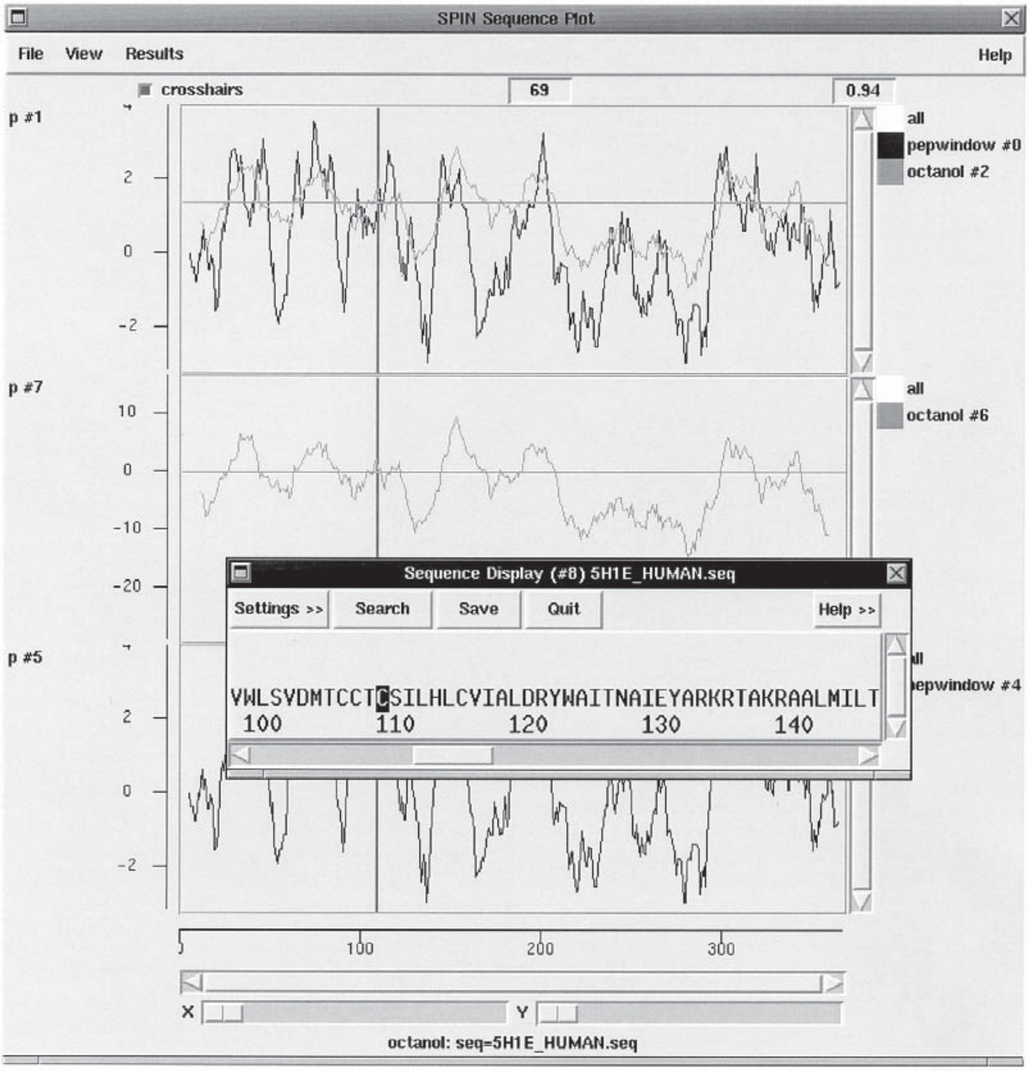
produce results that will be automatically loaded into SPIN, but other EMBOSS formats can be selected, for example, postscript output can be produced. Clicking the **OK** button will run cpgplot and the results will appear in a SPIN Sequence Plot.

An Example of SPIN EMBOSS Result

Figure 10 shows a SPIN Sequence Plot for two EMBOSS protein hydrophobicity analysis programs **octanol** and **pepwindow** applied to the SWISSPROT entry 5H1E_HUMAN, which is a member of family 1 of the G-protein coupled receptors. Here both programs have been applied twice and the results combined into a single Sequence Plot. The bottom two plots, partially obscured by the Sequence Display, are the individual results, and the top plot shows the two results superimposed. This superposition is achieved using SPIN's drag and drop functionality, that enables the user to compare the two analysis, a task which would be less straightforward without a graphical user interface.

A Summary of the Current EMBOSS Applications

Although many applications could be included in several categories, below we have divided the EMBOSS applications into separate lists: 1) Nucleic acid specific, 2) Protein specific, 3) Codons/Translation, 4) Alignment/Searching, and 5) Restriction Enzymes. Programs dealing with databases and those covering areas unrelated to sequences are not included.



```
VWLSVDMTCCTCSILHLCVIALDRYWAITNAIEYARKRTAKRAALMILT
 100       110       120       130       140
```

octanol: seq=5H1E_HUMAN.seq

Fig. 10. A SPIN Sequence Plot for EMBOSS programs octanol and pepwindow.

EMBOSS Nucleic Acid Specific Programs

Find the linguistic complexity in nucleotide sequences.
Count words of a specified size in a DNA sequence.
Calculate the fractional GC content of nucleic acid sequences.
Plot/report CpG rich areas.
Plot isochores in large DNA sequences.
Plot melting temperatures for DNA.
Create a chaos plot for a sequence.
Look for inverted/tandem repeats in a nucleotide sequence.
Nucleic acid pattern search.
Regular expression search of a nucleotide sequence.

Scan DNA sequences for transcription factors.
Find MAR/SAR sites in nucleic sequences.
Find nucleic acid binding domains.
Plot bending and curvature in B-DNA.
Calculate the twisting in a B-DNA sequence.
Do wobble base plot.

EMBOSS Protein Specific Programs

Protein charge plot.
Protein proteolytic enzyme or reagent cleavage digest.
Protein identification by mass spectrometry.
Protein pattern search.
Predict protein secondary structure.
Display protein hydropathy.
Hydrophobic moment calculation.
Match a PROSITE motif against a Protein Sequence Database.
Compare a protein sequence to the PROSITE motif database.
Predict transmembrane proteins.
Predict coiled coil regions.
Predict signal peptide cleavage sites.
Plot simple amino acid properties in parallel.
Protein helical net plot.
Show protein sequences as helices.
Regular expression search of a protein sequence.
Calculate the isoelectric point of a protein.
Find antigenic sites in proteins.
Find protein sequence regions with a biased composition.
Back translate a protein sequence.

EMBOSS Codons/Translation Programs

ORF property statistics.
Codon usage statistics.
Create a codon usage table.
Codon usage table comparison.
Extract CDS, mRNA and translations from feature tables.
Find and extract open reading frames.
Do synonymous codon usage Gribskov statistic plot.

EMBOSS Alignment/Searching Programs

DNA Sequence Comparison Plot.
Local/global alignment of two sequences.
Plot the quality of conservation of a sequence alignment.
Display aligned sequences, with coloring and boxing.
Create a consensus from multiple alignments.
Find differences (SNPs) between nearly identical sequences.
Display a non-overlapping wordmatch dotplot of two sequences.
Align EST and genomic DNA sequences.
Do an all-against-all comparison of a set of sequences.
Display a multiple sequence alignment.
Find all exact matches of a given size between 2 sequences.

Scan a sequence or database with a matrix or profile.
Gapped alignment for profiles.
Select primers for PCR and DNA amplification.
Search DNA sequences for matches with primer pairs.

EMBOSS Restriction Enzyme Analysis Programs

Display a sequence with e.g., restriction enzyme cut sites, translation.
Find Restriction Enzyme Cleavage Sites.
Silent mutation restriction enzyme scan.
Find and remove restriction sites but maintain the same translation.
Find restriction enzymes that produce a specific overhang.
Draw circular maps of DNA constructs.
Draw linear maps of DNA constructs.

Concluding Remarks

At the time of writing (July, 2001) both SPIN and EMBOSS are quite new, and the interface between them even newer. By the time this book appears we expect them to have matured into a cohesive, comprehensive and easy to use combination.

Glossary and Abbreviations

CREATE_EMBOSS_FILES A computer program for creating a weight matrix from a set of sequence alignments.

Emboss A suite of programs for comparing and analyzing sequences.

Global Alignment A full-length alignment between a pair of sequences.

Local Alignment An alignment between segments of two sequences.

Score Matrix A table of values assigning a score for aligning each possible pair of bases or amino acids.

SPIN A computer program for comparing and analyzing DNA sequences.

Weight Matrix A table of values assigning a score for each character type at each position in a DNA or protein motif.

Suggested Readings

Sequence Assembly

Bonfield, J. K., Smith, K. F., and Staden, R. (1995) A new DNA sequence assembly program, Nucleic Acids Res. 23, 4992–4999.

The SPIN Analytical Functions

Detecting Signals in Sequences

Fickett, J. W. (1982) Recognition of protein coding regions in DNA sequences, Nucleic Acids Res. 10, 5303–5318.

Staden, R. (1984) Computer methods to locate signals in nucleic acid sequences, Nucleic Acids Res. 12, 505–519.

Staden, R. (1984) Measurements of the effects that coding for a protein has on a DNA sequence and their use for finding genes, Nucleic Acids Res. 12, 551–567.

Staden, R. (1980) A program to search for tRNA genes, Nucleic Acids Res. 8, 817–825.

Sequence Alignment and Significance

Huang, X. Q. and Miller, W. A. (1991) Time-Efficient, Linear-Space Local Similarity Algorithm, Adv. Appl. Math. 12, 337–357.

Huang, X. (1994) On global sequence alignment, CABIOS 10, 227–235.

Staden, R. (1989) Methods for calculating the probabilities of finding patterns in sequences, Comput. Applic. Biosci. 5, 89–96.

Staden, R. (1982) An interactive graphics program for comparing and aligning nucleic acid and protein sequences, Nucleic Acids Res. 9, 2951–2961.

B. The Genome Database: *Analysis and Similarity Searching*

25 Ensembl

An Open-Source Software Tool for Large-Scale Genome Analysis

James W. Stalker and Anthony V. Cox

Introduction

The completion of the human genome project (HGP) by the Human Genome Consortium provides research and computational biologists with the first *working draft* sequence of the human genome. The sequence consists of approximately three billion base-pairs of DNA sequence information, the sheer scale of which presents immense computational problems.

Biological sequence analysis is recognized as presenting computational challenges as great as those faced in existing *big science* projects such as particle physics and space exploration. Conventional methods of genome data curation and annotation cannot keep up with the rate at which data are accumulating. The gap will continue to widen. Ensembl was designed to meet the need for an automated annotation system capable of handling large volumes of finished and unfinished sequence data.

Ensembl, a joint project of the Sanger Center and the European Bioinformatics Institute, provides researchers with access to state-of-the-art automated annotation of genomic data. By its nature, this annotation is complete and consistent across the genome. Ensembl will be a foundation for the next generation of sequence databases that provide a curated, distributed, and nonredundant view of model organism genomes. Initial Ensembl development was concentrated on human genomic data, however the model is now applied to other organisms, including mouse, fly and worm.

Ensembl: An Open-Source Tool

One of the major successes of the HGP was to make the human genome sequence freely available to all. However, the computational resources necessary to analyze the data are not widely available outside of large private companies. This could have had the effect of confining the usefulness of the raw genome data to a small number of well-funded groups. Ensembl has aimed to bring these data to the widest possible audience.

A central tenet of the Ensembl philosophy is openness: all data, software, and associated information is freely available to all and without restriction. The project has enthusiastically embraced the open-source ethic, with all code released under an open-source licence, and contributors around the globe supplying expertise and time free of charge.

The Ensembl Analysis System

Ensembl consists of two main parts: 1) the analysis pipeline, which adds new data and analyses to the core database, and 2) the API (application programming interface), which gives structured access to these data. The website (*see* Website: http://www.ensembl.org) is simply one implementation of a graphical interface to the core database through the API. Other implementations, such as java clients, stand-alone applications, and so forth, are certainly possible and indeed encouraged.

The Analysis Pipeline

The Ensembl analysis pipeline is fed raw sequence data and subjects it to a battery of analysis programs. As the data passes through the pipeline these analyses produce supporting evidence for the gene predictions. At first all annotations are merely *in silico* predictions, with subsequent steps sorting predictions into the two groups of known genes and novel predictions. The result is a database containing DNA sequence, predicted features on that sequence and a complete body of evidence supporting these predictions. The database contains only the results of the pipeline analysis as no features are imported from external databases. Ensembl *known* genes therefore are simply those predicted genes that have high similarity to genes confirmed by experimental evidence.

The API

The Ensembl API provides a representation of the data in an Ensembl database in terms of model biological objects (genes, clones, contigs, etc.). This abstraction shields programmers from the underlying complexity of the data, making it easier to retrieve information in a meaningful form. This makes the API an extremely powerful tool for biologists. It is simple, for example, to extract a specific clone object from the database and query it for such biological properties as contigs, length, sequence, and any features added by the analysis pipeline such as genes, repeats, and CpG islands.

The API allows for the attachment of external databases to the Ensembl core data. In this way, a wide range of additional information, such as maps, gene expression data, and sequence features, can be superimposed onto the Ensembl predictions. The initial implementation of the API is in Perl, built upon a layer of Bio-Perl objects. Other implementations and language bindings such as Java and Python are under development.

Ensembl Data Concepts

The Ensembl data model revolves around two central ideas: a *golden path* and *virtual contigs*. Genomic sequence data is accumulated as a set of overlapping clones, each containing one or more sequence fragments called *contigs*. In *unfinished* data, the order and orientation of these contigs is often unknown. *Finished* data has been fully assembled, such that each clone comprises only a single contig, i.e., the clone has been fully sequenced. The clone overlap information, combined with data from genetic maps, can be used to assemble the clones into a continuous pathway along a chromosome. This pathway through the data, containing the nonredundant sequence, is often referred to as *the golden path*.

It is often desirable to be able to work with regions of an assembly that do not map exactly to a single clone or contig, for example, one might wish to examine the first megabase of a chromosome. The Ensembl API allows this manipulation of arbitrary regions by providing an abstraction of the underlying golden path assembly, called a *virtual contig*. A virtual contig behaves as if it was a real contig, but may actually be constructed behind-the-scenes from multiple real (or *raw*) contigs and fragments of contigs.

Virtual contigs smooth over the underlying complexity of the fragmentary, redundant nature of the assembly, and allow you to treat the data as one continuous sequence. Much of the power and flexibility of the Ensembl API comes from being able to manipulate virtual contigs. For example, virtual contigs provide a *handle* for accessing features, such as genes, that span contig or clone boundaries.

Ensembl Website

The Ensembl website (*see* Website: http://www.ensembl.org), is an interactive graphical interface to the Ensembl database, using Perl scripts that use the API (*see* Subheading "Using the Ensembl API"). In keeping with the Ensembl philosophy of openness, all source codes for the website can be freely downloaded, so you can install your own local copy, or just look at how it works.

The website offers a hierarchical interface to the human genome, so that a user can *drill down* into the data by selecting successively more detailed views. For example, clicking on a chromosome in **MapView** takes you to a high-level display of the region in ContigView, and from there to detailed displays of individual clones, genes, transcripts, proteins, and other sequence features. As well as drilling into the data, you can search for particular identifiers, or find data using alignment methods such as BLAST and SSAHA. The website also provides extensive facilities for exporting data in a variety of forms, including FASTA, EMBL, tab-delimited lists, and several image formats.

DAS (Distributed Annotation System)

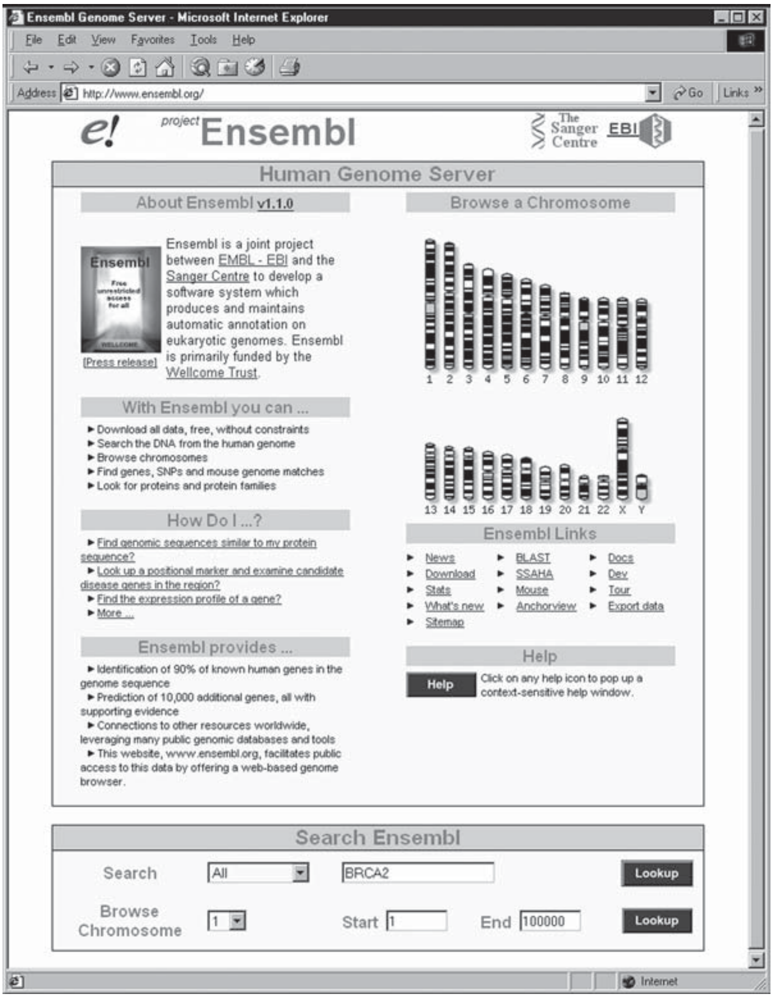
With the rapid proliferation of databases of genome data from many research institutions around the world, it is becoming a Sisyphean task to keep up to date with the flood of data. Rather than each institution trying to maintain local copies of all data, with all the attendant issues of synchronisation and duplication, it makes sense for institutions to concentrate on their own areas of expertise and then share this with everyone else. This is the foundation of the Distributed Annotation System (DAS). DAS provides a distributed network of genome sequence servers that provide a consistent reference *backbone* upon which anyone can layer their own annotation data. This annotation data can, in turn, be served up via DAS to client applications worldwide.

The Ensembl website allows the users to extend the site by dynamically adding information provided via remote DAS servers. For example, a ContigView display served from the Ensembl site in the UK can easily be configured to display additional annotation information provided by a DAS server in the US. This enables users to mix-and-match their displays to suit their research needs.

In return, the Ensembl golden path data is served as a reference sequence via DAS (*see* Website: http://das.ensembl.org).

Website Tour

As a quick introduction to the Ensembl website we will investigate the BRCA2 gene. We will find the gene, look at its details, examine the region it occupies in the



Ensembl is a joint project between EMBL - EBI and the Sanger Centre to develop a software system which produces and maintains automatic annotation on eukaryotic genomes. Ensembl is primarily funded by the Wellcome Trust.

With Ensembl you can ...

- Download all data, free, without constraints
- Search the DNA from the human genome
- Browse chromosomes
- Find genes, SNPs and mouse genome matches
- Look for proteins and protein families

How Do I ...?

- Find genomic sequences similar to my protein sequence?
- Look up a positional marker and examine candidate disease genes in the region?
- Find the expression profile of a gene?
- More ...

Ensembl provides ...

- Identification of 90% of known human genes in the genome sequence
- Prediction of 10,000 additional genes, all with supporting evidence
- Connections to other resources worldwide, leveraging many public genomic databases and tools
- This website, www.ensembl.org, facilitates public access to this data by offering a web-based genome browser.

Click on any help icon to pop up a context-sensitive help window.

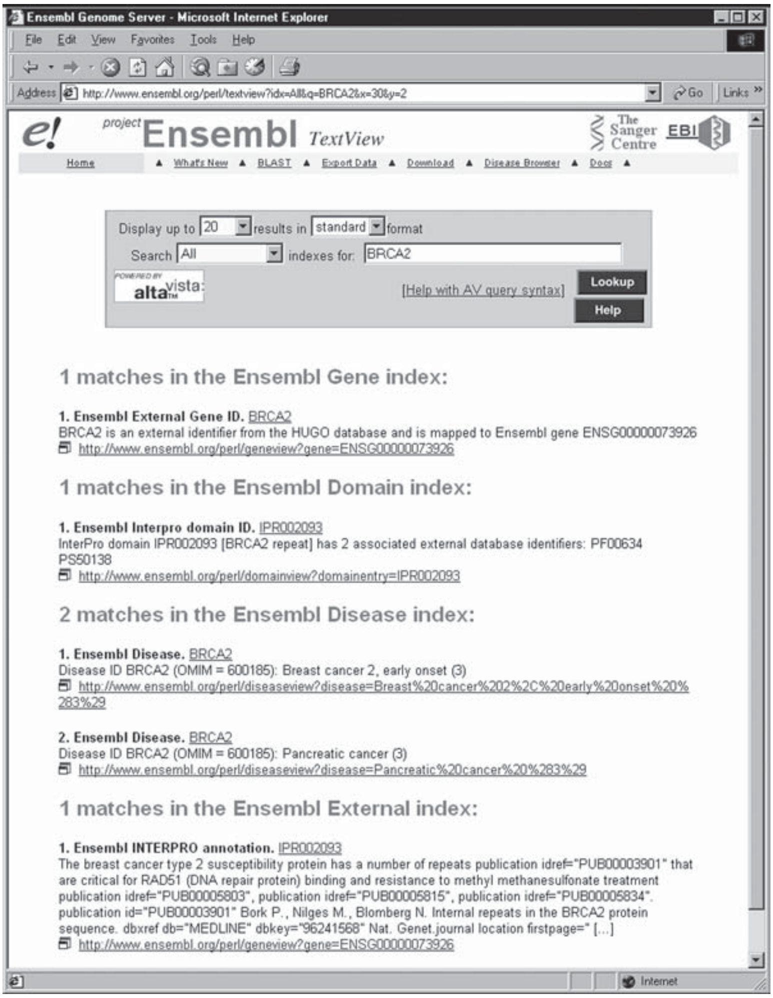
Fig. 1. The Ensembl Website home page.

genome, and then export some reference data. These are all tasks you can do in a script, of course, but the website puts an easy-to-use interface on top of all the code as shown in Fig. 1.

Searching for a Feature

The website uses the AltaVista search engine to provide a full index of the databases, enabling very fast and comprehensive searches. We will use this to find the BRCA2 gene by name.

Enter **BRCA2** into the search box on the home page. You can change the Search Type to limit the search to a particular type of feature, in this case gene, but we will leave it set to **All** to see everything that the database holds about the BRCA2 identifier.


Ensembl Genome Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.ensembl.org/perl/textview?idx=All&q=BRCA2&x=30&y=2
Go Links
e! project Ensembl TextView
The Sanger Centre EBI
Home ▲ What's New ▲ BLAST ▲ Export Data ▲ Download ▲ Disease Browser ▲ Docs ▲
powered by altavista
Lookup
Help
Internet

Display up to 20 results in standard format

Search All indexes for: BRCA2

[Help with AV query syntax]

1 matches in the Ensembl Gene index:

1. Ensembl External Gene ID. BRCA2
BRCA2 is an external identifier from the HUGO database and is mapped to Ensembl gene ENSG00000073926
http://www.ensembl.org/perl/geneview?gene=ENSG00000073926

1 matches in the Ensembl Domain index:

1. Ensembl Interpro domain ID. IPR002093
InterPro domain IPR002093 [BRCA2 repeat] has 2 associated external database identifiers: PF00634 PS50138
http://www.ensembl.org/perl/domainview?domainentry=IPR002093

2 matches in the Ensembl Disease index:

1. Ensembl Disease. BRCA2
Disease ID BRCA2 (OMIM = 600185): Breast cancer 2, early onset (3)
http://www.ensembl.org/perl/diseaseview?disease=Breast%20cancer%202%2C%20early%20onset%20%283%29

2. Ensembl Disease. BRCA2
Disease ID BRCA2 (OMIM = 600185): Pancreatic cancer (3)
http://www.ensembl.org/perl/diseaseview?disease=Pancreatic%20cancer%20%283%29

1 matches in the Ensembl External index:

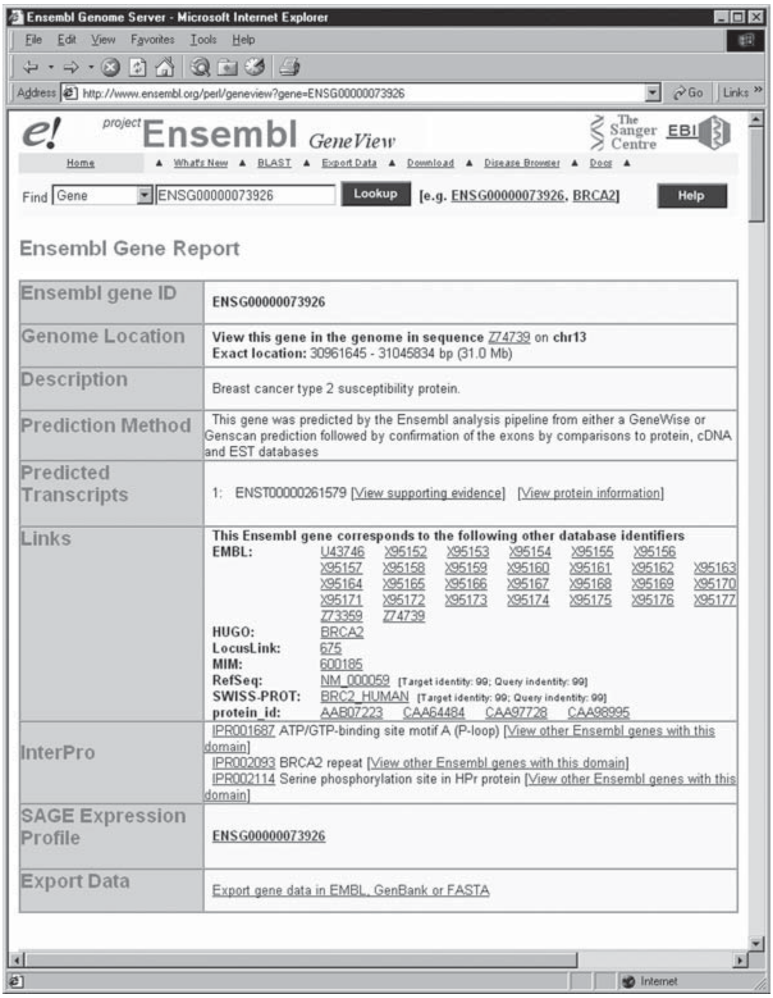
1. Ensembl INTERPRO annotation. IPR002093
The breast cancer type 2 susceptibility protein has a number of repeats publication idref="PUB00003901" that are critical for RAD51 (DNA repair protein) binding and resistance to methyl methanesulfonate treatment publication idref="PUB00005803", publication idref="PUB00005815", publication idref="PUB00005834". publication id="PUB00003901" Bork P., Nilges M., Blomberg N. Internal repeats in the BRCA2 protein sequence. dbxref db="MEDLINE" dbkey="96241568" Nat. Genet.journal location firstpage=" [...]
http://www.ensembl.org/perl/geneview?gene=ENSG00000073926

Fig. 2. Results of the search for BRCA2.

You should see more than one match for BRCA2, as Fig. 2. shows. The search currently matches the gene BRCA2, an InterPro protein domain, diseases associated with the gene, and also finds BRCA2 in some external annotation documentation. For now we are only interested in the gene, so click on the link from the match to the gene index. This will take us to a detailed view of the gene in GeneView.

GeneView

The GeneView page contains much detail and information about BRCA2 including its Ensembl identifier, description, DBLinks, and InterPro protein domain matches. This is shown in Fig. 3. Further down the page is the sequence of the corresponding transcripts, exons, and splice sites.


Ensembl Genome Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.ensembl.org/perl/geneview?gene=ENSG00000073926
Go
Links »
e! project Ensembl GeneView
The Sanger Centre
EBI
Home ▲ What's New ▲ BLAST ▲ Export Data ▲ Download ▲ Disease Browser ▲ Docs ▲
Find Gene ENSG00000073926
Lookup
[e.g. ENSG00000073926, BRCA2]
Help
Internet

Ensembl Gene Report

Ensembl gene ID	**ENSG00000073926**
Genome Location	**View this gene in the genome in sequence** Z74739 on **chr13** **Exact location:** 30961645 - 31045834 bp (31.0 Mb)
Description	Breast cancer type 2 susceptibility protein.
Prediction Method	This gene was predicted by the Ensembl analysis pipeline from either a GeneWise or Genscan prediction followed by confirmation of the exons by comparisons to protein, cDNA and EST databases
Predicted Transcripts	1: ENST00000261579 [View supporting evidence] [View protein information]
Links	**This Ensembl gene corresponds to the following other database identifiers** **EMBL:** U43746 X95152 X95153 X95154 X95155 X95156 X95157 X95158 X95159 X95160 X95161 X95162 X95163 X95164 X95165 X95166 X95167 X95168 X95169 X95170 X95171 X95172 X95173 X95174 X95175 X95176 X95177 Z73359 Z74739 **HUGO:** BRCA2 **LocusLink:** 675 **MIM:** 600185 **RefSeq:** NM_000059 [Target identity: 99; Query indentity: 99] **SWISS-PROT:** BRC2_HUMAN [Target identity: 99; Query indentity: 99] **protein_id:** AAB07223 CAA64484 CAA97728 CAA98995
InterPro	IPR001687 ATP/GTP-binding site motif A (P-loop) [View other Ensembl genes with this domain] IPR002093 BRCA2 repeat [View other Ensembl genes with this domain] IPR002114 Serine phosphorylation site in HPr protein [View other Ensembl genes with this domain]
SAGE Expression Profile	**ENSG00000073926**
Export Data	Export gene data in EMBL, GenBank or FASTA

Fig. 3. The gene summary section of GeneView for BRCA2.

For each transcript, GeneView has diagrams showing the exon structure. These can be seen for a transcript of BRCA2 in Fig. 4. If we want to investigate the genomic location of the gene in more detail, there is a section in the gene summary called Genome Location. This specifies the clone, chromosome and base-pair coordinates where the gene is located. This information is linked into the detailed genome browser page using **ContigView**.

ContigView

ContigView is the main genome browser display for the Ensembl website. It allows the user to *walk* freely up and down any chromosome, and displays all the sequence features appropriate to the current region. There are a variety of navigation

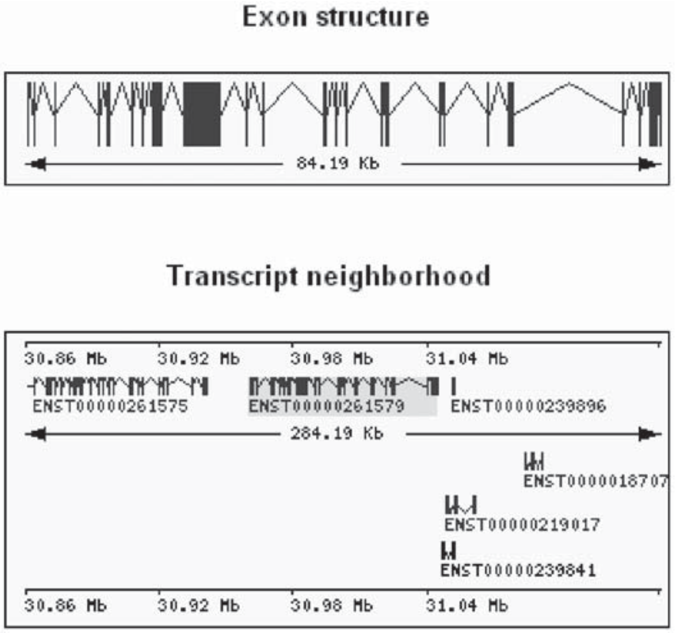


Fig. 4. Exon structure and transcript locale diagrams from GeneView.

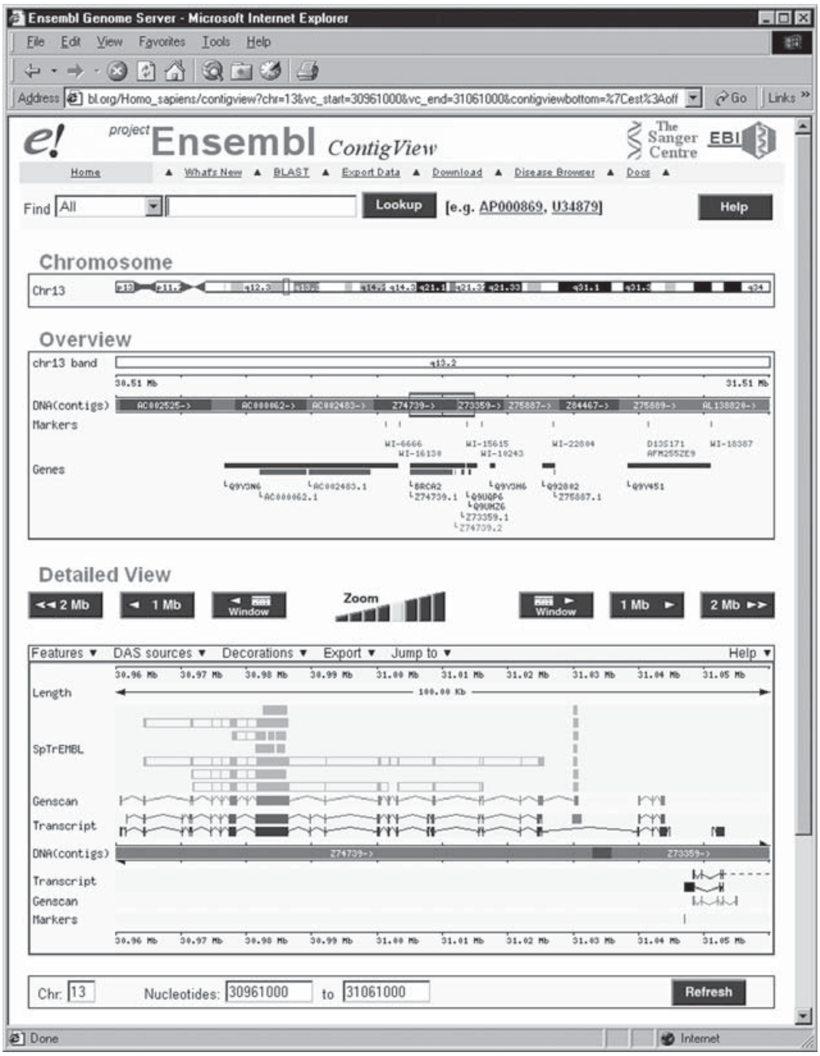
tools provided to let the user move to any location, zoom in and out, and focus on a specific area.

The ContigView display is customizable. The user can toggle the display of any feature type, change how they are drawn, and even change their color. ContigView displays also show any DAS sources that the user has added. The page therefore can be customized precisely to show only the data needed for the task the user is working on.

Selecting any of the features by moving the mouse over the object on ContigView will display a drop-down menu. These menus contain more feature information, and links that jump to a detailed view of the specific feature.

As you can see from Fig. 5, ContigView shows a hierarchy of views. The topmost view shows the chromosome that we are looking at. A box surrounds the region that is represented in the next image: the overview display. The overview shows, by default, one megabase of DNA, and is a high-level view of the region. For this reason, the overview only shows landmark features such as markers and genes. Known genes, novel genes, curated EMBL genes, and pseudogenes are distinctively colored. Clicking anywhere on the overview will focus the detailed view on that point and this region of focus is displayed again.

The bottom view is the detailed display. It shows a relatively short region of the genome, but can display all the features present in that area. Only a few feature sets are turned on in Fig. 5, but you can see the large dark BRCA2 transcript in the center, and see that this prediction is supported by both EMBL and Genscan transcripts. SpTrEMBL protein homology matches are shown above these.



Address: bl.org/Homo_sapiens/contigview?chr=13&vc_start=30961000&vc_end=31061000&contigviewbottom=%7Cest%3Aoff

Find: All

Chr: 13 Nucleotides: 30961000 to 31061000

Fig. 5. ContigView centered around the BRCA2 gene.

The thick bar along the middle of the detailed display is a representation of the contigs in the golden path at that point. Features above the contigs are on the forward strand of the DNA at that point (as is our transcript), while those below are on the reverse strand.

The menus at the top of the detailed display provide additional customization options. The **Features** menu has checkboxes for the user to toggle the display of the different feature types. **Decorations** gives the same control over items on the display such as the length bar or a %GC plot. **DAS sources** is for toggling configured DAS sources, and also contains links for adding and removing sources. The **Jump to** menu lets you visit the equivalent region in the UCSC genome browser. Finally, the **Export** menu, as shown in Fig. 6, gives access to ExportView.

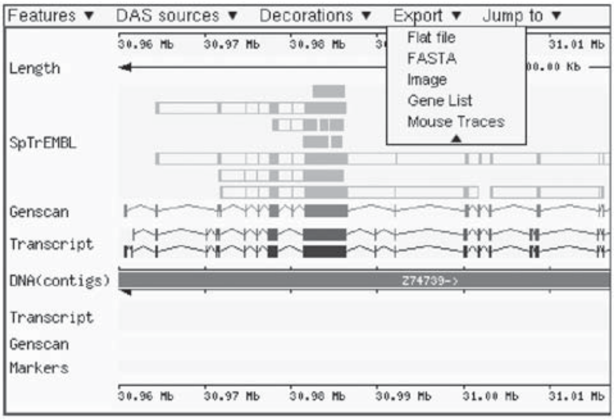


Fig. 6. The Export menu on the ContigView detailed display.

ExportView

ExportView is a tool for exporting data from a feature or region in a variety of formats. The page shows a tabbed dialog, with a different tab for each export format. At the time of writing, these were, as illustrated in Fig. 7, Flat File, FASTA, Image, Gene List, and Traces.

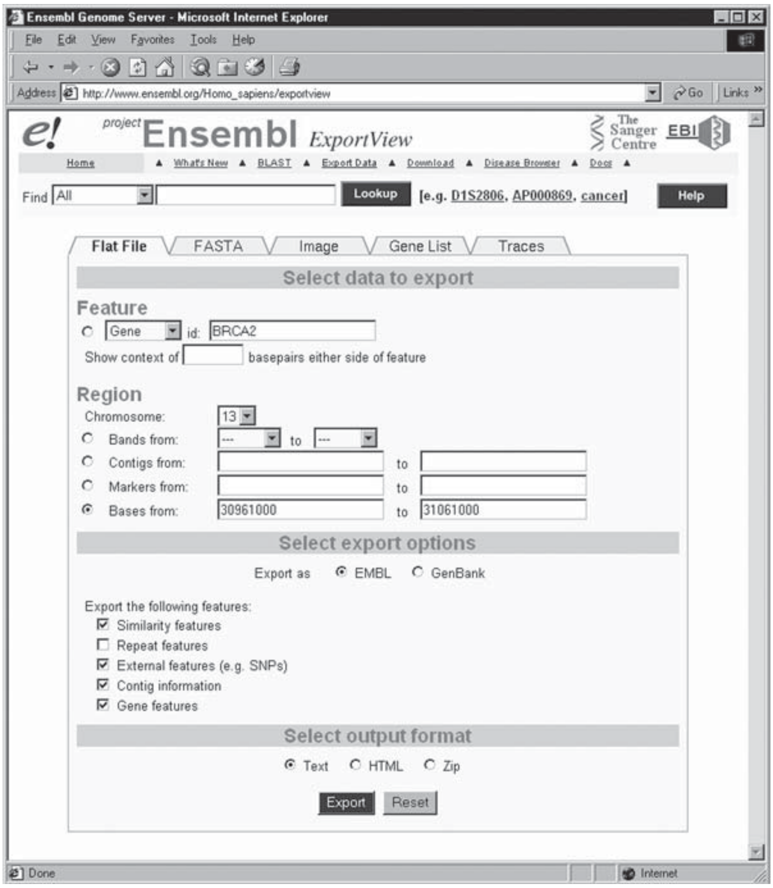
The **Flat File**, **FASTA**, **Gene List**, and **Traces** tabs all allow export of data as text (useful for cut and paste of data), HTML (the same data as text, but with links back into the website), or as a zipped text file for download. The Flat File tab allows you to export a feature or region in EMBL or GenBank format and the FASTA tab provides FASTA format.

Image exporting enables you to create a detailed image, similar to that shown at the bottom of ContigView, of any region of the genome. Images can be exported in GIF/PNG, PostScript, Scaleable Vector Graphics (SVG), and Windows MetaFile (WMF) formats. The Trace tab gives access to raw sequence traces that have been matched to the human genome, for example mouse traces.

The Gene List tab is extremely useful. It exports tab-delimited data that can be easily imported into other applications (such as spreadsheets) for further analysis. For example, if we click on the **Gene List** tab after coming from our BRCA2 region in contigview, the base-pair location fields will be filled in with the appropriate coordinates. The rest of the tab contains a large list of features we can include in or exclude from the export. As an example, check the **Ensembl gene id**, **known gene external id**, **chromosome start**, **chromosome end**, and **chromosome name** boxes. Click **export**, and the results should look like:

```
# gene_name          external_id      chrom_start      chrom_end        chrom_band

ENSG00000073926      BRCA2            30961645         31045834         q13.2
ENSG00000102832      Q9UQP6           31047843         31062857         q13.2
ENSG00000120702      Q9UHZ6           31052055         31053944         q13.2
```

This is a quick and powerful way to extract a lot of useful information from the Ensembl database.


Ensembl Genome Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address
Go
Links
e! project Ensembl ExportView
The Sanger Centre
EBI
Home ▲ What's New ▲ BLAST ▲ Export Data ▲ Download ▲ Disease Browser ▲ Docs ▲
Lookup
Help
Flat File
FASTA
Image
Gene List
Traces
Export
Reset
Done
Internet

http://www.ensembl.org/Homo_sapiens/exportview

Find All [e.g. D1S2806, AP000869, cancer]

Select data to export

Feature

○ Gene id: BRCA2

Show context of basepairs either side of feature

Region

Chromosome: 13

○ Bands from: --- to ---

○ Contigs from: to

○ Markers from: to

● Bases from: 30961000 to 31061000

Select export options

Export as ● EMBL ○ GenBank

Export the following features:

☑ Similarity features

□ Repeat features

☑ External features (e.g. SNPs)

☑ Contig information

☑ Gene features

Select output format

● Text ○ HTML ○ Zip

Fig. 7. ExportView.

Using the Ensembl API

Installation

Before using the Ensembl API, you must first obtain and install the Ensembl software modules. The components you will require for a basic installation are:

- Bioperl: Provides the foundation objects for Ensembl.
- Ensembl: The core functionality and database objects.

These components and instructions for retrieving and installing them, can be found on the Ensembl website. You will also need an Ensembl database to run your code. Again, the website provides instructions on obtaining and installing a freely available copy of the databases. If you do not have or do not want to install a local Ensembl database, you can direct your software to a publicly available copy at the Sanger Center. This is the database we will be using for these examples.

Finally, you will need a local installation of the Perl programming language, which is free to download (*see* Website: http://www.perl.com).

Set Up of the Ensembl Software Environment

Once you have installed Ensembl and Bioperl you need to configure your environment so that your Perl script will know where the modules are. The easiest way to do this is to add the location of the modules to the PERL5LIB environment variable.

For example, if you installed Ensembl and Bioperl into a directory called */home/bob*, then set the environment as follows:

In UNIX:

```
csh:    setenv PERL5LIB /home/bob/ensembl/modules:/home/bob/bioperl
bash:   export PERL5LIB = /home/bob/ensembl/modules:/home/bob/bioperl
```

On Windows:

```
        set PERL5LIB c:\home\bob\ensembl\modules;c\home\bob\bioperl
```

Worked Examples

The following examples are intended to provide a brief introduction to using the Ensembl API for manipulating data in an Ensembl database. The code used here only scratches the surface; the API has many powerful features. Additional details are available at the website, which always contains the latest API documentation (*see* Website: http://www.ensembl.org).

Working with Clones and Contigs

As previously discussed (*see* Subheading “Ensembl Data Concepts”), we learned that a clone is a fundamental unit of a sequenced genome and is made up of one or more contigs. We are going to use the Ensembl API to retrieve a clone from the database, extract the contigs that make up that clone, and then look at their sequences.

Connecting to the Database

We will assume that you do not have a local Ensembl database installed and instead connect to a publicly available copy of the Ensembl databases on kaka.sanger.ac.uk. Of course, this is far slower than accessing a local copy, which is a must for any serious use.

We use a DBAdaptor module from the API to connect to the database:

```
use Bio::EnsEMBL::DBSQL::DBAdaptor;
```

To make the connection, we need to specify the host machine that is serving the database (kaka.sanger.ac.uk), the name of the database to connect to (the current database at time of writing is *ensembl110*), and the username to connect as (kaka accepts the user *anonymous*).

The code to make the connection is:

```
my $database = Bio::EnsEMBL::DBSQL::DBAdaptor->new( -host  =>‘kaka.sanger.ac.uk’,
                                                    -user => ‘anonymous’,
                                                    -dbname => ‘ensembl110’ );
```

If the connection fails, you will get an error message. Otherwise, you are free to start working with the data in the database.

Fetching a Clone from the Database

It is very simple to retrieve a clone by its accession number by typing:

```
my $clone = $database->get_Clone(‘AP000869’);
```

The variable $clone now contains an Ensembl clone object.

You can check it is the clone that you asked for by printing its identifier:

```
print “Clone id is “. $clone->id . “\n”;
```

Getting the Contigs from a Clone

Now that we have a clone object, we can get the contigs that make up that clone:

```
my @contigs = $clone->get_all_Contigs;
```

@contigs is now an array that holds a list of all the contigs in the clone. We can loop over these contigs and print out some information about them:

```
foreach my $contig (@contigs) {
    print "Contig ". $contig->id . " is ". $contig->length ." base-pairs long\n";
}
```

Getting the Sequence from a Contig

The method call *primary_seq* retrieves a sequence object from a contig. To get the DNA sequence from the contig, you ask the sequence object for its sequence string:

```
my $seqobj = $contig->primary_seq;
print $seqobj->seq . "\n";
```

This will print the correct sequence, but it is not exactly pretty output. Something like FASTA format would be much more useful. We could of course do this ourselves manually, but Bioperl already provides a way to do this via the *SeqIO* object.

```
use Bio::SeqIO;
my $seqio = Bio::SeqIO->new(-format => 'fasta');
$seqio->write_seq($seqobj);
```

Finished Example

Our completed script looks like this:

```
#!/usr/local/bin/perl

use strict;
use Bio::EnsEMBL::DBSQL::DBAdaptor;
use Bio::SeqIO;


my $database = Bio::EnsEMBL::DBSQL::DBAdaptor->new( -host => 'kaka.sanger.ac.uk',
                                                    -user=> 'anonymous',
                                                    -dbname=>'ensembl110');

my $clone = $database->get_Clone('AP000869');
print "Clone id is ". $clone->id . "\n";


my @contigs = $clone->get_all_Contigs;


foreach my $contig (@contigs){
    print "Contig ". $contig->id . " is ". $contig->length ." base-pairs long\n";
}


foreach my $contig (@contigs){
    my $seqio = Bio::SeqIO->new (-format => 'fasta');
    my $seqobj = $contig->primary_seq;
    $seqio->write_seq($seqobj);
}
```

To recap: this script connects to an Ensembl database, retrieves a clone object from the database by its identifier, collects all the contigs in that clone and prints their sequences in FASTA format.

The output should look something like this:

```
Clone id is AP000869
Contig AP000869.2.154917.155980 is 1064 base-pairs long
Contig AP000869.2.27859.38788 is 10930 base-pairs long
Contig AP000869.2.38889.50392 is 11504 base-pairs long
Contig AP000869.2.158624.159840 is 1217 base-pairs long
.
.
etc
```

Followed by the FASTA sequence of the contigs:

```
>AP000869.2.154917.155980
AGGGAAAAAAAAGTGATGGAGAAATGAGAACTTTCACCAACCCACCCAAATTGAGAGAAT
CTGTAGACCTGTCTTGCCAGAAGTATTTAAAAAAAAAACTTCTTCAGAGAAAAGAAAGAT
GATATAAGTTAGAAACTTTGAACTACATAAACAAGGGAAGAGCTTTAGAGAAGGAATAAG
.
.
etc
```

Working with a Virtual Contig

A virtual contig acts like the *raw* contigs that we have just been using. The calls we made on a contig will also work just as well on a virtual contig. Because genes are built on virtual contigs in the analysis pipeline, and because many genes span more than one contig, it is preferable (and often essential) to use virtual contigs to access them. In this example, we will build a virtual contig and look at the genes it contains.

Connecting to the Database

We connect to the database in exactly the same way as in the previous example, i.e., via a DBAdaptor. However, we are working with virtual rather than raw contigs and need access to the golden path (remember that a virtual contig is just a chunk of this continuous assembly). We access the golden path through another adaptor object — the *static golden path adaptor*. It is called static because the golden path has been pregenerated and stored in the database, rather than being generated dynamically when required (this is quite possible, but rather slow). The golden path we use in Ensembl is currently provided by UCSC. Ensembl has the capacity to store and use multiple different golden paths; first, the type of golden path is relayed to the DBAdaptor by typing:

```
$database->static_golden_path_type(‘UCSC’);
```

and then get the static golden path adaptor:

```
my $sa = $database->get_StaticGoldenPathAdaptor;
```

We can now use the static golden path adaptor to assemble virtual contigs, much as we used the DBAdaptor to retrieve the raw contigs. In fact, we could get the same region as before, using the call:

```
my $vcontig = $sa->fetch_VirtualContig_of_clone(‘AP000869’);
```

Building a Virtual Contig

There are several calls on the static golden path adaptor that retrieve virtual contigs. This can be queried by clone, contig, gene, and transcript identifiers. Probably the most interesting is to use the contiguous nature of the golden path to fetch an arbitrary region of sequence that might span multiple clones:

```
My $vcontig = $sa->fetch_VirtualContig_by_chr_start_end('chrX',1,500000);
```

$vcontig now contains a virtual contig built from the first half megabase of chromosome X. Even though it is constructed of many underlying contigs stitched together, it acts as though it were a single Ensembl contig, for example, we can extract the DNA sequence using the same method we used for a raw contig:

```
print $vcontig->primary_seq->seq;
```

With just this small snippet of code, we are already able to access the DNA sequence of any arbitrary region of a chromosome, as much or as little as is required.

Virtual Contig Properties

Because a virtual contig is created from a chromosomal assembly, it contains information about its absolute position in that chromosome. For example:

```
print "This virtual contig is from chromosome ". $vcontig->_chr_name. "\n";
print "Base-pairs ".$vcontig->_global_start. " to ". $vcontig->_global_end. "\n";
```

This information may not appear very useful when implemented on a virtual contig constructed by chromosomal position, but the clone identifier could use this to locate that clone within the golden path.

We can also examine the underlying structure of the virtual contig, by retrieving the contigs that comprise the contig. The call to do this is **each_MapContig**, and it gives us a list of contig objects, in a similar way in the previous example that the **get_all_Contigs** call on the clone did:

```
my @contigs = $vcontig->_vmap->each_MapContig;
print "Virtual Contig ".$vcontig->id. " is made up of the following contigs: \n";
foreach my $mapcontig (@contigs){
    print $mapcontig->contig->id."\n";
}
```

Retrieving Genes

We can access all the genes on a virtual (or raw) contig with:

```
my @genes = $vcontig->get_all_Genes;
```

As you should be coming to expect, this presents an array of gene objects. We can print their Ensembl identifiers with:

```
foreach my $gene (@genes){
    print "Gene ". $gene->id . "\n";
}
```

Ensembl identifiers do not yield a bounty of information about a gene. They are automated predictions of genes from the analysis pipeline, not experimentally confirmed genes. However, we can tell if an Ensembl gene has been matched to a known gene by checking the **is_known** property of the gene. If the gene is known, then we can look at the list of matches for that gene, and perhaps retrieve a more familiar name. The list is actually a list of matches against databases of known genes, and these connections between predictions and known genes are called *DBLinks*. In addition to the DBLink information, we can retrieve a description of known genes from the relevant SwissProt or RefSeq entry. For example:

```
if ($gene->is_known){
     print "Gene ". $gene->id. " : ". $gene->description."\n";
     foreach my $link ($gene->each_DBLink){
    print " Links:". $link->display_id. " from ". $link->database. "\n";
     }
}
else {
     print "Gene ". $gene->id. " is a novel prediction.\n";
}
```

Transcripts from Genes

Just as gene objects are retrieved from contig objects, transcript objects are retrieved from genes. A gene has one or more transcripts; one for each alternatively spliced mRNA. You can view the transcripts from a gene as follows:

```
my @transcripts = $gene->each_Transcript;
```

As you would expect, you can call the id method of a transcript object to retrieve its Ensembl identifier. Like Ensembl gene identifiers, these do not mean very much in themselves. Transcripts are most interesting for their structure and what they encode.

Exons from Transcripts

Like stacking Russian dolls, we can again extract more objects from the objects we already have. Transcripts are made up of exons, which can be pulled out with the familiar syntax:

```
my @exons = $transcript->each_Exon; }
```

We can query exons for their sequence and coordinates as:

```
foreach my $exon (@exons){
    print $exon->id. " : ". $exon->start ." - ". $exon->end. "\n";
    print $exon->seq->seq ."\n";
}
```

Notice again that the first call on the exon for that sequence retrieves a sequence object, this can be followed by a DNA string by asking for the sequence from the object.

Transcript Translations

The final manipulation of transcripts yields the peptide sequence from the translated transcript. The Bio::SeqIO module is used since it yields a nicely formatted output. We can retrieve the peptide object from the transcript by asking for its translation:

```
my $seqio = Bio::SeqIO->new (-format => 'fasta');
my $peptide = $transcript->translate;
$seqio->write_seq($peptide);
```

Finished Example

Our completed script looks like this:

```
#!/usr/local/bin/perl

use strict;
use Bio::EnsEMBL::DBSQL::DBAdaptor;
use Bio::SeqIO;

my $database = Bio::EnsEMBL::DBSQL::DBAdaptor->new(-host => ‘kaka.sanger.ac.uk’,
                                                   -user=> ‘anonymous’,
                                                   -dbname=>’ensembl110');

$database->static_golden_path_type(‘UCSC’);
my $sa = $database->get_StaticGoldenPathAdaptor;

my $vcontig = $sa->fetch_VirtualContig_by_chr_start_end(‘chrX’,1,500000);

print “Virtual contig: “. $vcontig->_chr_name. “ base-pairs “;
print $vcontig->_global_start. “ to “. $vcontig->_global_end. “\n”;

print “is built from: \n”;
foreach my $mapcontig ($vcontig->_vmap->each_MapContig){
    print “Raw contig: “.$mapcontig->contig->id.”\n”;
}

print “\n and contains the following genes: \n”;
my @genes = $vcontig->get_all_Genes;

foreach my $gene(@genes){
    if ($gene->is_known){
        print “Gene “. $gene->id. “ : “. $gene->description.”\n”;
        foreach my $link ($gene->each_DBLink){
            print “ Links:”. $link->display_id. “ from “. $link->database.”\n”;
        }
        print “\n”;
    }
    else {
        print “Gene “. $gene->id. “ is a novel prediction.\n”;
    }
}
print “\n”;

foreach my $gene(@genes){
    print “Gene: “. $gene->id.”\n”;
    foreach my $transcript ($gene->each_Transcript){
        print “Transcript: “.$transcript->id.”\n”;
        print “Exons: \n”;
        foreach my $exon ($transcript->each_Exon){
            print “\t”.$exon->id. “ : “. $exon->start. “ - “. $exon->end. “\n”;
        }
        print “Peptide:\n”;
        my $seqio = Bio::SeqIO->new(-format => ‘fasta’);
        my $peptide = $transcript->translate;
        $seqio->write_seq($peptide);
        print “\n”;
    }
}
```

This script connects to an ensembl database, retrieves a static golden path adaptor, and uses that to build a virtual contig of an arbitrary region of chromosome X. It then retrieves all the genes from the virtual contig, all the transcripts from the genes, and all the exons from the transcripts. Finally we print out the peptide translations of the transcripts in FASTA format are printed. As shown previously, the Ensembl API enables you to extract and manipulate biologically meaningful objects from the Ensembl database with a surprisingly small amount of code.

Future Developments

Ensembl is under active development. The already rich API is growing constantly to provide convenient access to a wider range of genome data resources. Each release of the database sees the inclusion of additional data of ever-increasing quality. The website is also being adapted to represent these underlying improvements in the API and data, and to meet the increasingly complex demands of the research community. Future developments currently being considered include:

- Extended support for DAS
- Cross-species data linking and comparison
- Multiple language bindings for the API, including Java & Python
- Remote object services via CORBA and SOAP

Contact List

The Ensembl project has a very active development mailing list called *ensembl-dev*. All of the Ensembl team participate in the list, so if you have a question or comment about Ensembl you can be sure it will be answered there. There is also a low traffic mailing list, ensembl-announce, which is used to announce major updates or a new release. To subscribe to these lists, send an email to: majordomo@ebi.ac.uk with either *subscribe ensembl-announce* or *subscribe ensembl-dev* in the body of the mail. These mailing lists are archived on the Ensembl website.

Ensembl also maintains a helpdesk facility that will answer any questions pertaining to the project. Either send an email to: helpdesk@ensembl.org, or use the form on the website.

The Ensembl web team is always interested in any feedback you might have on the site. Comments and suggestions are all carefully considered, and often make their way into the next version of the site. The web team can be contacted through the Ensembl helpdesk, or via webmaster@ensembl.org.

Suggested Readings

Ensembl: An Open-Source Tool

Raymond, E. S. (1999) The Cathedral and the Bazaar.
see Websites: http://www.ensembl.org/Download/ and http://www.ftp.ensembl.org
http://www.bioperl.org/
http://www.open-bio.org/

Ensembl Website

see Websites: http://www.sanger.ac.uk/Software/analysis/SSAHA/
http://www.biodas.org/

26 The PIR for Functional Genomics and Proteomics

Cathy H. Wu

Introduction

The human genome project has revolutionized the practice of biology and the future potential of medicine. Complete genomes continue to be sequenced *en masse*. Meanwhile, there is growing recognition that proteomic studies bring the researcher closer to the actual biology than studies of gene sequence or gene expression alone. High-throughput studies are being conducted and rapid advances being made in areas such as protein expression, protein structure and function, and protein-protein interactions. Given the enormous increase in genomic, proteomic, and molecular data, computational approaches, in combination with empirical methods, are expected to become essential for deriving and evaluating hypotheses. To fully explore these valuable data, advanced bioinformatics infrastructures must be developed for biological knowledge management. One major challenge lies in the volume, complexity, and dynamic nature of the data, which are being collected and maintained in heterogeneous and distributed sources. New approaches need to be devised for data collection, maintenance, dissemination, query, and analysis. The Protein Information Resource (PIR) aims to serve as an integrated public resource of functional annotation of proteins to support genomic/proteomic research and scientific discovery.

The PIR was established in 1984 as a resource to assist researchers in the identification and interpretation of protein sequence information. The PIR, along with the Munich Information Center for Protein Sequences (MIPS) and the Japan International Protein Information Database (JIPID), continues to enhance and distribute the PIR-International Protein Sequence Database. The database evolved from the first comprehensive collection of macromolecular sequences in the *Atlas of Protein Sequence and Structure* published from 1965–1978 under the editorship of Margaret O. Dayhoff, who pioneered molecular evolution research.

PIR provides many protein databases and data analysis tools employing a family classification approach to facilitate protein annotation and data integration. The PIR-International Protein Sequence Database (PSD) is the major annotated protein database in the public domain, containing about 283,000 sequences covering the entire taxonomic range. The PIR superfamily organization allows complete and nonoverlapping clustering of all proteins. Comprehensive protein information is available from iProClass, which includes family classification at the superfamily, domain, and motif levels; structural and functional features of proteins; as well as cross-references to over 50 databases of protein families, structures, functions, genes, genomes, literature, and taxonomy. The nonredundant reference protein database, PIR-NREF, provides timely and comprehensive data collected from PIR-PSD, Swiss-Prot, TrEMBL, GenPept, RefSeq, and PDB, with composite protein names and literature information for about 1,000,000 proteins. To promote database interoperability and improve annotation quality, we have adopted controlled vocabulary, standard nomenclature, and common ontologies, and distinguish experimentally determined from predicted protein features. We also provide open and modular database architecture and XML data distribution to assist data integration in a distributed networking environment. The PIR databases are implemented in the object-relational database system and are freely accessible from our web site (*see* Website: http://pir.georgetown.edu/). The site features data mining and sequence analysis tools for information retrieval and functional identification of proteins based on both sequence and annotation information. It supports the exploration of proteins and comparative studies of various family relationships. Such knowledge is crucial to our understanding of protein evolution, structure and function, and is important for functional genomic and proteomic research.

PIR-International Protein Sequence Database

The PIR-International Protein Sequence Database (PSD) is a nonredundant, expertly annotated, fully classified, and extensively cross-referenced protein sequence database in the public domain. Release 73.0, July 2002, contained about 283,000 protein sequences with comprehensive coverage across the entire taxonomic range, including sequences from publicly available complete genomes.

Superfamily Classification

A unique characteristic of the PIR-PSD is the superfamily classification that provides complete and nonoverlapping clustering of proteins based on global (end-to-end) sequence similarity. Sequences in the same superfamily share common domain architecture (i.e., have the same number, order and types of domains), and do not differ excessively in overall length unless they are fragments or result from alternate splicing or initiators. The automated classification system places new members into existing superfamilies and defines new superfamily clusters using parameters including the percentage of sequence identity, overlap length ratio, distance to neighboring superfamily clusters, and overall domain arrangement. Currently, over 99% of the sequences are classified into families of closely related sequences (at least 45% identical) and over two thirds of sequences are classified into >36,200 superfamilies. The automated classification is being augmented by manual curation of superfamilies, starting with those containing at least one definable domain, to provide superfamily names, brief descriptions, bibliography, list of representative and seed members, as well as domain and motif architecture characteristic of the superfamily. Sequences in PIR-PSD are also classified with homology domains and sequence motifs. Homology domains, which are shared by more than one superfamily, may constitute evolutionary building blocks, while sequence motifs represent functional sites or conserved regions.

The classification allows the systematic detection of genome annotation errors based on comprehensive superfamily and domain classification. Several annotation

See companion CD for color Fig. 1

errors originated from different genome centers have lead to the *transitive catastrophe*. Figure 1 illustrates an example where several members of three related superfamilies were originally misannotated because only local domain (not global superfamily) relationships were considered.

The classification also provides the basis for rule-based procedures that are used to propagate information-rich annotations among similar sequences and to perform integrity checks. These scripts use the superfamily/family classification system, sequence patterns and profiles to produce highly specific annotations. False-positives are avoided by applying automated annotations only to classified members of the families and superfamilies for which the annotation has been validated. Integrity checks are based on PIR controlled vocabulary, standard nomenclature (such as IUBMB Enzyme Nomenclature) and a thesaurus of synonyms or alternate names.

Evidence Attribution and Bibliography Submission

Attribution of protein annotations to validated experimental sources provides an effective means to avoid propagating errors that may have resulted from large-scale genome annotation. To distinguish experimentally verified from computationally predicted data, PIR-PSD entries are labeled with status tags such as *validated* or *similarity* in protein Title, Function and Complex annotations, as well as tags such as *experimental*, *predicted*, *absent*, or *atypical* in Feature annotations. The validated Function or Complex annotation includes hypertext-linked PubMed unique identifiers for the articles in which the experimental determination are reported.

Linking protein data to more literature data that describes or characterizes the proteins is crucial to increasing the amount of experimental information and improving the quality of protein annotation. We have developed a bibliography submission system for the scientific community to submit, categorize, and retrieve literature information for PSD protein entries. The submission interface guides users through steps in mapping the paper citation to given protein entries, entering the literature data, and summarizing the literature data using categories such as genetics, tissue/cellular localization, molecular complex or interaction, function, regulation, and disease. Also included is a literature information page that provides literature data mining and displays both references cited in PIR and submitted by users.

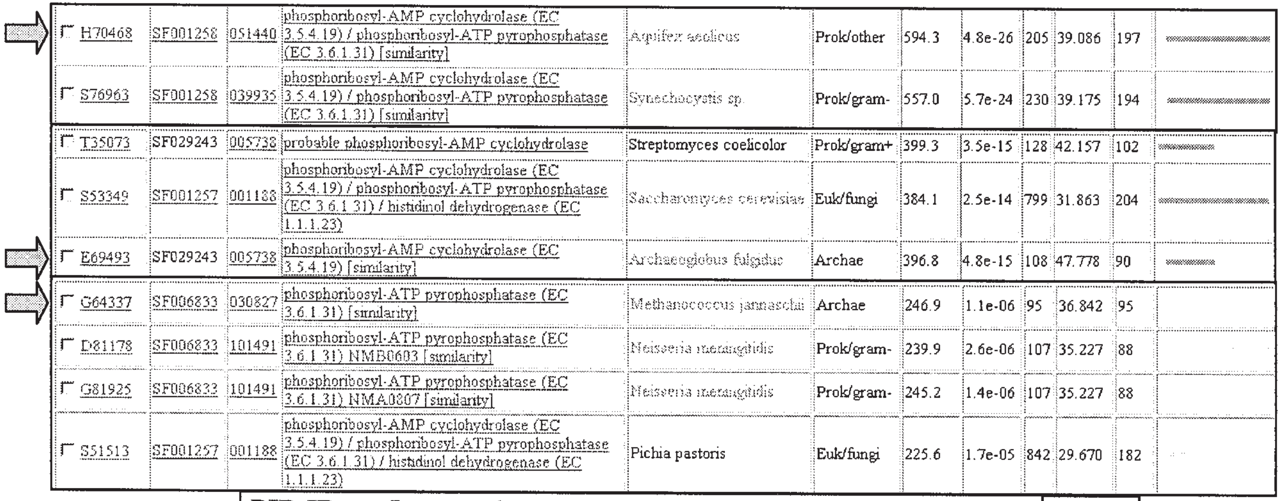
See companion CD for color Fig. 2

Integrated Protein Classification Database

The iProClass (Integrated Protein Classification) database (*see* Fig. 2), is designed to provide comprehensive descriptions of all proteins and to serve as a framework for data integration in a distributed networking environment. The protein information includes family relationships at both global (superfamily/family) and local (domain, motif, site) levels, as well as structural and functional classifications and features of proteins. A modular architecture organizes the information into multiple database components for Sequence, Superfamily, Domain, Motif, Structure, and Function.

See companion CD for color Fig. 3

The current version (Release 2.4, August 2002) consists of about 810,000 nonredundant PIR-PSD, Swiss-Prot, and TrEMBL proteins organized with more than 36,200 PIR superfamilies; 145,340 families; 3840 PIR homology and Pfam domains; 1300 ProClass/ProSite motifs; 280 PIR post-translational modification sites; and links to over 50 databases of protein families, structures, functions, genes, genomes, literature, and taxonomy. The Sequence report (*see* Fig. 3) provides detailed protein



□ H70468	SF001258	051440	phosphoribosyl-AMP cyclohydrolase (EC 3.5.4.19) / phosphoribosyl-ATP pyrophosphatase (EC 3.6.1.31) [similarity]	Aquifex aeolicus	Prok/other	594.3	4.8e-26	205	39.086	197	▬
□ S76963	SF001258	039935	phosphoribosyl-AMP cyclohydrolase (EC 3.5.4.19) / phosphoribosyl-ATP pyrophosphatase (EC 3.6.1.31) [similarity]	Synechocystis sp.	Prok/gram-	557.0	5.7e-24	230	39.175	194	▬
□ T35073	SF029243	005738	probable phosphoribosyl-AMP cyclohydrolase	Streptomyces coelicolor	Prok/gram+	399.3	3.5e-15	128	42.157	102	▬
□ S53349	SF001257	001188	phosphoribosyl-AMP cyclohydrolase (EC 3.5.4.19) / phosphoribosyl-ATP pyrophosphatase (EC 3.6.1.31) / histidinol dehydrogenase (EC 1.1.1.23)	Saccharomyces cerevisiae	Euk/fungi	384.1	2.5e-14	799	31.863	204	▬
□ E69493	SF029243	005738	phosphoribosyl-AMP cyclohydrolase (EC 3.5.4.19) [similarity]	Archaeoglobus fulgidus	Archae	396.8	4.8e-15	108	47.778	90	▬
□ G64337	SF006833	030827	phosphoribosyl-ATP pyrophosphatase (EC 3.6.1.31) [similarity]	Methanococcus jannaschii	Archae	246.9	1.1e-06	95	36.842	95	
□ D81178	SF006833	101491	phosphoribosyl-ATP pyrophosphatase (EC 3.6.1.31) NMB0603 [similarity]	Neisseria meningitidis	Prok/gram-	239.9	2.6e-06	107	35.227	88	
□ G81925	SF006833	101491	phosphoribosyl-ATP pyrophosphatase (EC 3.6.1.31) NMA0807 [similarity]	Neisseria meningitidis	Prok/gram-	245.2	1.4e-06	107	35.227	88	
□ S51513	SF001257	001188	phosphoribosyl-AMP cyclohydrolase (EC 3.5.4.19) / phosphoribosyl-ATP pyrophosphatase (EC 3.6.1.31) / histidinol dehydrogenase (EC 1.1.1.23)	Pichia pastoris	Euk/fungi	225.6	1.7e-05	842	29.670	182	

PIR ID	Imported	Corrected	Superfamily
H70468	3.6.1.31	3.5.4.19/3.6.1.31	SF001258
E69493	3.5.4.19/3.6.1.31	3.5.4.19	SF029243
G64337	3.5.4.19	3.6.1.31	SF006833

A

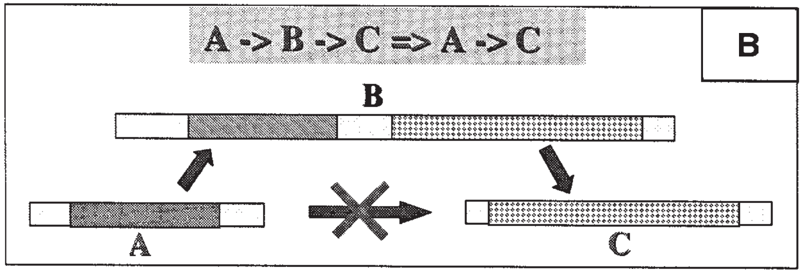


Fig. 1. Genome sequence annotation: transitive catastrophe. **(A)** Misannotation of three imported sequence entries is later corrected based on superfamily classification. **(B)** Transitive identification error, illustrated in entry G64337, often involves multi-domain proteins.

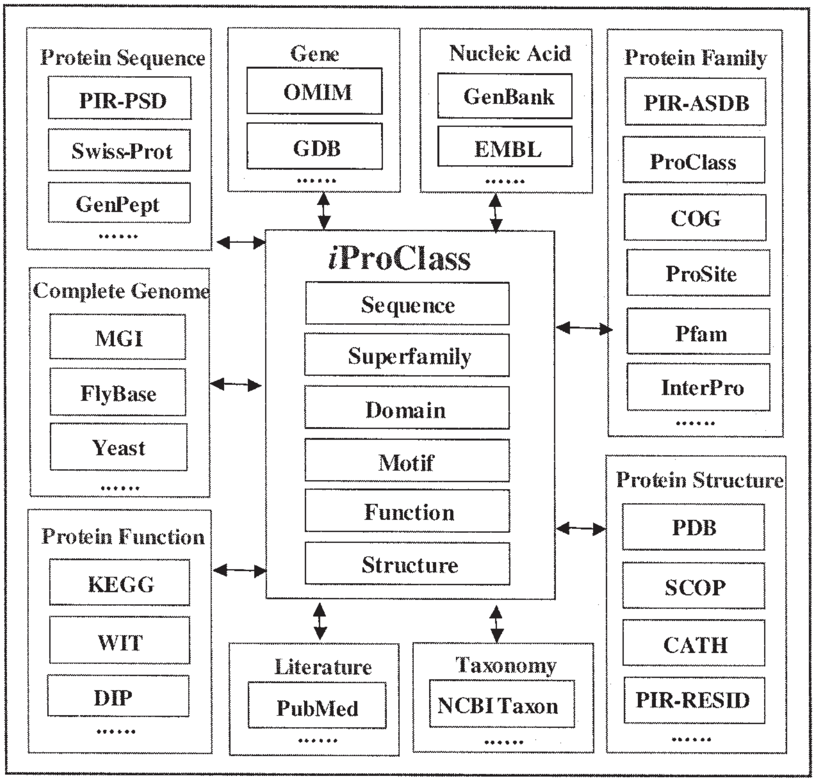
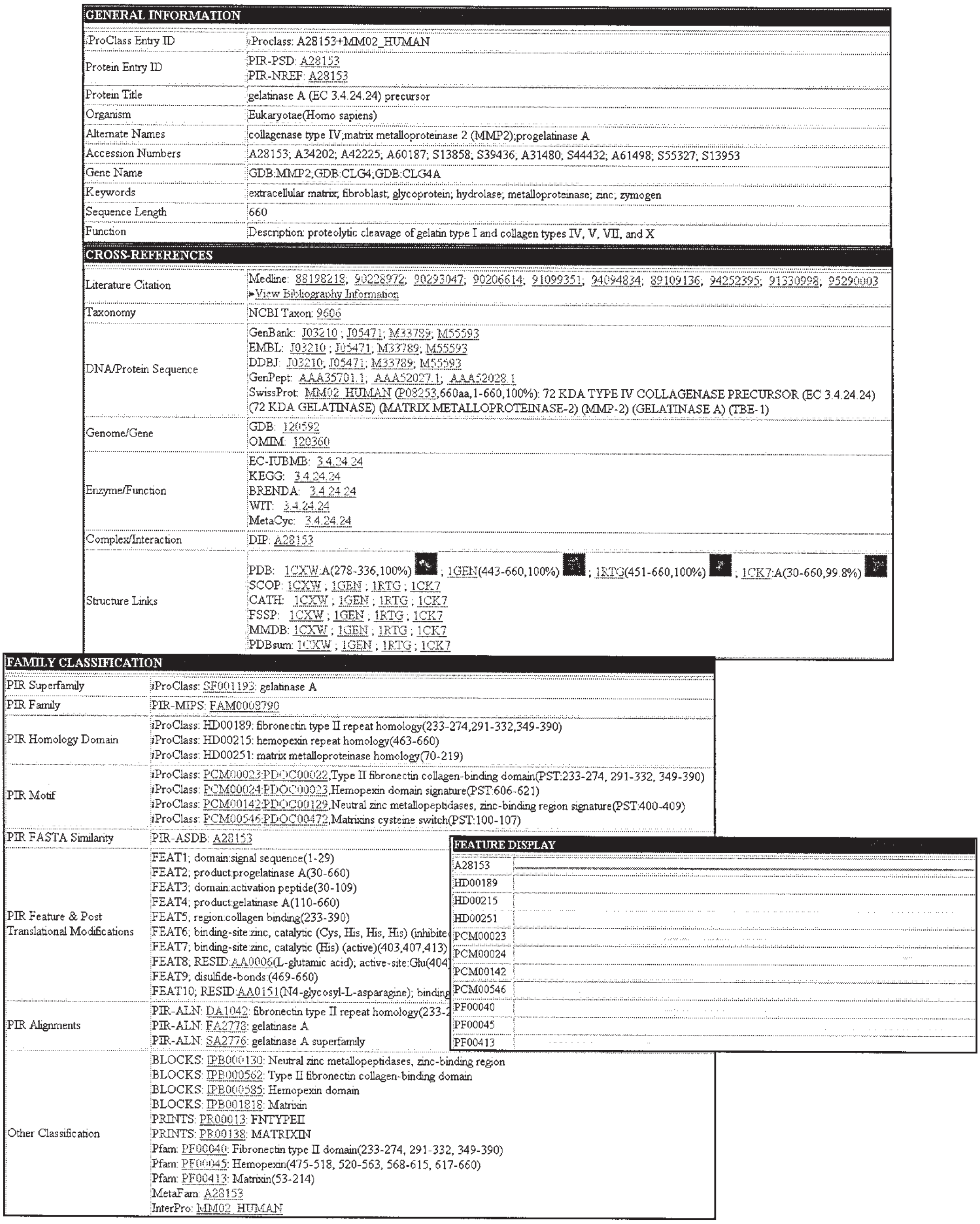


Fig. 2. iProClass database for data integration: modular database components and extensive links to molecular biology databases.

summary information in four sections: General Information, Cross-References, Family Classification, and Feature Display, with extensive hypertext links for further exploration and graphical display of domain and motif regions. The Superfamily report provides summaries including membership information with length, taxonomy, and keyword statistics; complete member listing separated by major kingdoms, family relationships, and structure and function cross-references.

Directly linked to the iProClass sequence report are two additional PIR databases, Annotation and Similarity DataBase (ASDB) and post-translational modification database (RESID). PIR-ASDB lists pre-computed, biweekly-updated FASTA sequence similarity researched neighbors of all PSD sequences with annotation information and graphical displays of sequence similarity matches. PIR-RESID documents over 280 post-translational modifications and links to PSD entries containing either experimentally determined or computationally predicted modifications with evidence tags.

To be implemented are the Domain-Motif components that represent domain and motif-centric views with direct mapping to superfamilies, the Function component that describes functional properties of enzymes and other activities and relationships such as families, pathways, and processes, as well as the Structure component that describes structural properties and relate structural classes to evolution and func-



GENERAL INFORMATION	
iProClass Entry ID	iProclass: A28153+MM02_HUMAN
Protein Entry ID	PIR-PSD: A28153 PIR-NREF: A28153
Protein Title	gelatinase A (EC 3.4.24.24) precursor
Organism	Eukaryotae(Homo sapiens)
Alternate Names	collagenase type IV;matrix metalloproteinase 2 (MMP2);progelatinase A
Accession Numbers	A28153; A34202; A42225; A60187; S13858; S39436; A31480; S44432; A61498; S55327; S13953
Gene Name	GDB:MMP2;GDB:CLG4;GDB:CLG4A
Keywords	extracellular matrix; fibroblast; glycoprotein; hydrolase; metalloproteinase; zinc; zymogen
Sequence Length	660
Function	Description: proteolytic cleavage of gelatin type I and collagen types IV, V, VII, and X

CROSS-REFERENCES	
Literature Citation	Medline: 88198218; 90228972; 90293047; 90206614; 91099351; 94094834; 89109136; 94252395; 91330998; 95290003 ▸View Bibliography Information
Taxonomy	NCBI Taxon: 9606
DNA/Protein Sequence	GenBank: J03210 ; J05471; M33789; M55593 EMBL: J03210 ; J05471; M33789; M55593 DDBJ: J03210; J05471; M33789; M55593 GenPept: AAA35701.1; AAA52027.1; AAA52028.1 SwissProt: MM02_HUMAN (P08253,660aa,1-660,100%): 72 KDA TYPE IV COLLAGENASE PRECURSOR (EC 3.4.24.24) (72 KDA GELATINASE) (MATRIX METALLOPROTEINASE-2) (MMP-2) (GELATINASE A) (TBE-1)
Genome/Gene	GDB: 120592 OMIM: 120360
Enzyme/Function	EC-IUBMB: 3.4.24.24 KEGG: 3.4.24.24 BRENDA: 3.4.24.24 WIT: 3.4.24.24 MetaCyc: 3.4.24.24
Complex/Interaction	DIP: A28153
Structure Links	PDB: 1CXW:A(278-336,100%) ; 1GEN(443-660,100%) ; 1RTG(451-660,100%) ; 1CK7:A(30-660,99.8%) SCOP: 1CXW ; 1GEN ; 1RTG ; 1CK7 CATH: 1CXW ; 1GEN ; 1RTG ; 1CK7 FSSP: 1CXW ; 1GEN ; 1RTG ; 1CK7 MMDB: 1CXW ; 1GEN ; 1RTG ; 1CK7 PDBsum: 1CXW ; 1GEN ; 1RTG ; 1CK7

FAMILY CLASSIFICATION	
PIR Superfamily	iProClass: SF001193: gelatinase A
PIR Family	PIR-MIPS: FAM0008790
PIR Homology Domain	iProClass: HD00189: fibronectin type II repeat homology(233-274,291-332,349-390) iProClass: HD00215: hemopexin repeat homology(463-660) iProClass: HD00251: matrix metalloproteinase homology(70-219)
PIR Motif	iProClass: PCM00023:PDOC00022,Type II fibronectin collagen-binding domain(PST:233-274, 291-332, 349-390) iProClass: PCM00024:PDOC00023,Hemopexin domain signature(PST:606-621) iProClass: PCM00142:PDOC00129,Neutral zinc metallopeptidases, zinc-binding region signature(PST:400-409) iProClass: PCM00546:PDOC00472,Matrixins cysteine switch(PST:100-107)
PIR FASTA Similarity	PIR-ASDB: A28153
PIR Feature & Post Translational Modifications	FEAT1; domain:signal sequence(1-29) FEAT2; product:progelatinase A(30-660) FEAT3; domain:activation peptide(30-109) FEAT4; product:gelatinase A(110-660) FEAT5; region:collagen binding(233-390) FEAT6; binding-site:zinc, catalytic (Cys, His, His, His) (inhibite FEAT7; binding-site:zinc, catalytic (His) (active)(403,407,413) FEAT8; RESID:AA0006(L-glutamic acid); active-site:Glu(404) FEAT9; disulfide-bonds:(469-660) FEAT10; RESID:AA0151(N4-glycosyl-L-asparagine); binding
PIR Alignments	PIR-ALN: DA1042: fibronectin type II repeat homology(233-2 PIR-ALN: FA2778: gelatinase A PIR-ALN: SA2776: gelatinase A superfamily
Other Classification	BLOCKS: IPB000130: Neutral zinc metallopeptidases, zinc-binding region BLOCKS: IPB000562: Type II fibronectin collagen-binding domain BLOCKS: IPB000585: Hemopexin domain BLOCKS: IPB001818: Matrixin PRINTS: PR00013: FNTYPEII PRINTS: PR00138: MATRIXIN Pfam: PF00040: Fibronectin type II domain(233-274, 291-332, 349-390) Pfam: PF00045: Hemopexin(475-518, 520-563, 568-615, 617-660) Pfam: PF00413: Matrixin(53-214) MetaFam: A28153 InterPro: MM02_HUMAN

FEATURE DISPLAY	
A28153	
HD00189	
HD00215	
HD00251	
PCM00023	
PCM00024	
PCM00142	
PCM00546	
PF00040	
PF00045	
PF00413	

Fig. 3. iProClass protein sequence entry report (for a retrievable example *see* Website: http://pir.georgetown.edu/cgi-bin/iproclass/iproclass?choice=entry&id=A28153).

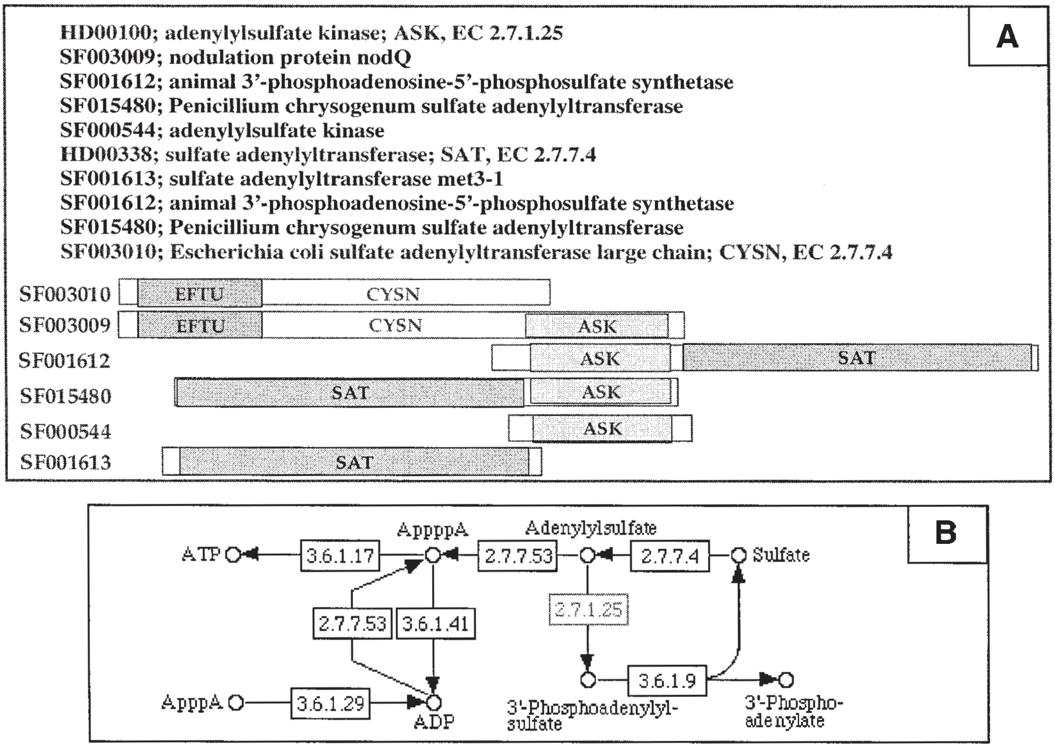

A
HD00100; adenylylsulfate kinase; ASK, EC 2.7.1.25
SF003009; nodulation protein nodQ
SF001612; animal 3'-phosphoadenosine-5'-phosphosulfate synthetase
SF015480; Penicillium chrysogenum sulfate adenylyltransferase
SF000544; adenylylsulfate kinase
HD00338; sulfate adenylyltransferase; SAT, EC 2.7.7.4
SF001613; sulfate adenylyltransferase met3-1
SF001612; animal 3'-phosphoadenosine-5'-phosphosulfate synthetase
SF015480; Penicillium chrysogenum sulfate adenylyltransferase
SF003010; Escherichia coli sulfate adenylyltransferase large chain; CYSN, EC 2.7.7.4
SF003010
EFTU
CYSN
SF003009
EFTU
CYSN
ASK
SF001612
ASK
SAT
SF015480
SAT
ASK
SF000544
ASK
SF001613
SAT
B
ATP
3.6.1.17
ApppA
2.7.7.53
Adenylylsulfate
2.7.7.4
Sulfate
2.7.7.53
3.6.1.41
2.7.1.25
3.6.1.9
ApppA
3.6.1.29
ADP
3'-Phosphoadenylyl-sulfate
3'-Phospho-adenylate

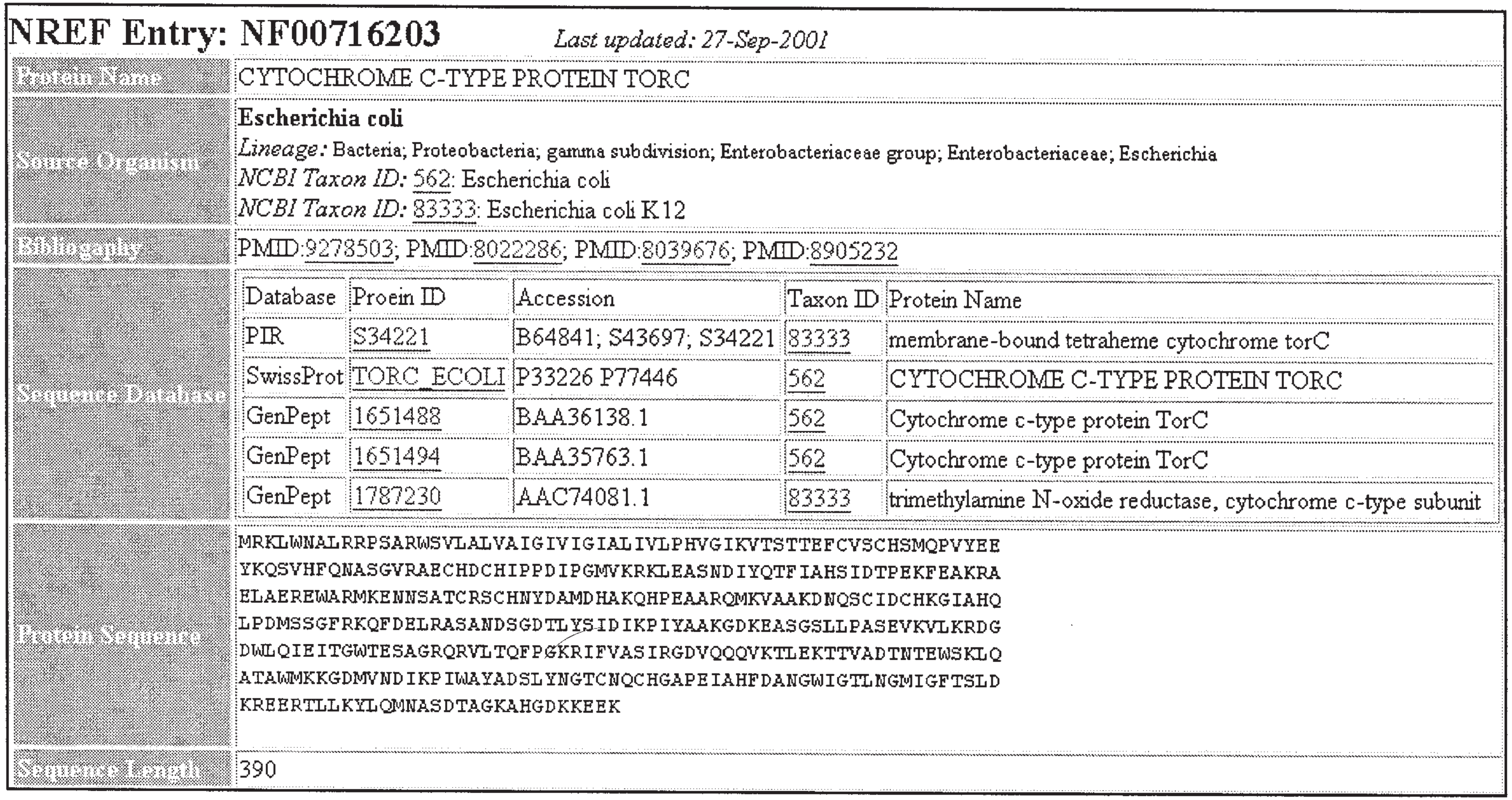
Fig. 4. Superfamily-domain-function relationship to reveal protein functional association beyond sequence homology. **(A)** Association of ASK (EC 2.7.1.25) and SAT/CYSN (EC2.7.7.4) in multi-domain proteins. **(B)** Association of ASK and SAT/CYSN in a metabolic pathway.

See companion CD for color Fig. 4

tion. Such data integration is important in revealing protein functional associations beyond sequence homology, as illustrated in the following example. As shown in Fig. 4A, the ASK domain (EC 2.7.1.25) appears in four different superfamilies, all having different overall domain arrangements. Except for SF000544, the other three superfamilies are bi-functional, all containing sulfate adenylyltrans-ferase (SAT) (EC 2.7.7.4). However, the same SAT enzymatic activity is found in two distinct sequence types, the SAT domain and CYSN homology. Furthermore, both EC 2.7.1.25 and EC 2.7.7.4 are in adjacent steps of the same metabolic pathway (*see* Fig. 4B). This example demonstrates that protein function may be revealed based on domain and/or pathway association, even without obvious sequence homology. The iProClass database design presents complex superfamily-domain-function relationships to assist functional identification or characterization of proteins.

PIR-Nonredundant Reference (PIR-NREF) Protein Database

As a major resource of protein information, one of our primary aims is to provide a timely and comprehensive collection of all protein sequence data that keeps pace with the genome sequencing projects and contains source attribution and minimal redundancy. The PIR-NREF protein database includes all sequences from PIR-PSD,



NREF Entry: NF00716203 *Last updated: 27-Sep-2001*

Protein Name	CYTOCHROME C-TYPE PROTEIN TORC
Source Organism	**Escherichia coli** *Lineage:* Bacteria; Proteobacteria; gamma subdivision; Enterobacteriaceae group; Enterobacteriaceae; Escherichia *NCBI Taxon ID:* 562: Escherichia coli *NCBI Taxon ID:* 83333: Escherichia coli K12
Bibliogaphy	PMID:9278503; PMID:8022286; PMID:8039676; PMID:8905232

Sequence Database

Database	Proein ID	Accession	Taxon ID	Protein Name
PIR	S34221	B64841; S43697; S34221	83333	membrane-bound tetraheme cytochrome torC
SwissProt	TORC_ECOLI	P33226 P77446	562	CYTOCHROME C-TYPE PROTEIN TORC
GenPept	1651488	BAA36138.1	562	Cytochrome c-type protein TorC
GenPept	1651494	BAA35763.1	562	Cytochrome c-type protein TorC
GenPept	1787230	AAC74081.1	83333	trimethylamine N-oxide reductase, cytochrome c-type subunit

Protein Sequence

```
MRKLWNALRRPSARWSVLALVAIGIVIGIALIVLPHVGIKVTSTTEFCVSCHSMQPVYEE
YKQSVHFQNASGVRAECHDCHIPPDIPGMVKRKLEASNDIYQTFIAHSIDTPEKFEAKRA
ELAEREWARMKENNSATCRSCHNYDAMDHAKQHPEAARQMKVAAKDNQSCIDCHKGIAHQ
LPDMSSGFRKQFDELRASANDSGDTLYSIDIKPIYAAKGDKEASGSLLPASEVKVLKRDG
DWLQIEITGWTESAGRQRVLTQFPGKRIFVASIRGDVQQQVKTLEKTTVADTNTEWSKLQ
ATAWMKKGDMVNDIKPIWAYADSLYNGTCNQCHGAPEIAHFDANGWIGTLNGMIGFTSLD
KREERTLLKYLQMNASDTAGKAHGDKKEEK
```

Sequence Length 390

Fig. 5. PIR-NREF nonredundant reference protein database for timely and comprehensive collection of protein data with source attribution.

Table 1
Major PIR Web Pages for Data Mining and Sequence Analysis

Description	*Web Page URL*
PIR Home	http://pir.georgetown.edu
PIR-PSD	~/pirwww/search/textpsd.shtml
iProClass	~/iproclass
PIR-NREF	~/pirwww/search/prinref.shtml
PIR-ASDB	~/cgi-bin/asdblist.pl?id=CCHU
Bibliography submission	~/pirwww/literature.html
List of PIR databases	~/pirwww/dbinfo/dbinfo.html
List of PIR search tools	~/pirwww/search/searchseq.html
FTP site	ftp://nbrfa.georgetown.edu/pir_databases

~ = http://pir.georgetown.edu

See companion CD for color Fig. 5

Swiss-Prot, TrEMBL, RefSeq, GenPept, PDB, and other protein databases. The NREF entries, each representing an identical amino acid sequence from the same source organism redundantly represented in one or more underlying protein databases, can serve as the basic unit for protein annotation. The National Center for Biotechnology Information (NCBI) taxonomy is used as the ontology for matching source organism names at the species or strain (if known) levels. The NREF report (*see* Fig. 5) provides source attribution (containing protein IDs, accession numbers, and protein names from underlying databases), in addition to taxonomy, amino acid sequence, and composite literature data. The composite protein names, including synonyms, alternate names and even misspellings, can be used to assist ontology development of protein names and the identification of misannotated proteins. Related sequences, including identical sequences from different organisms and closely related sequences within the same organism are also listed. The database presently consists of about 800,000 entries and is updated biweekly. Future versions of iProClass and PIR-ASDB will be based on the PIR-NREF database.

PIR System Distribution

PIR Web Site

The PIR web site (*see* Website: http://pir.georgetown.edu) connects data mining and sequence analysis tools to underlying databases for exploration of protein information and discovery of new knowledge. The site has been redesigned to include a user-friendly navigation system and more graphical interfaces and analysis tools. The PIR-PSD, iProClass, and NREF pages represent primary entry points in the PIR web site. A list of the major PIR pages is shown in Table 1.

The PIR-PSD interface provides entry retrieval, batch retrieval, basic or advanced text searches, and various sequence searches. The iProClass interface also includes both sequence and text searches. The BLAST search returns best-matched proteins and superfamilies, each displayed with a one-line summary linking to complete reports. Peptide match allows protein identification based on peptide sequences. Text search supports direct search of the underlying Oracle tables using unique identifiers or combinations of text strings, based on a Java program running JDBC. The NREF database is searchable for BLAST searching, peptide matching, and direct report retrieval based on the NREF ID or the entry identifiers of the source databases. Other sequence searches supported on the PIR web site include FASTA, pattern matching, hidden Markov model (HMM) domain and motif search, Smith-Waterman pair-wise alignment, ClustalW multiple alignment, and GeneFIND family identification.

PIR FTP Site

The PIR anonymous FTP site (*see* Website: ftp://nbrfa.georgetown.edu/pir_databases) provides direct file transfer. Files distributed include the PIR-PSD (quarterly release and interim updates), PIR-NREF, other auxiliary databases, other documents, files, and software programs. The PIR-PSD is distributed as flat files in NBRF and CODATA formats, with corresponding sequences in FASTA format. Both PIR-PSD and PIR-NREF are also distributed in XML format with the associated Document Type Definition (DTD) file.

The PIR-PSD, iProClass, and PIR-NREF databases have been implemented in Oracle 8i object-relational database system on our Unix server. To enable open source distribution, the databases have been mapped to MySQL and ported to Linux. To establish reciprocal links to PIR databases, to host a PIR mirror web site, or to request PIR database schema, please contact Website: pirmail@nbrf.georgetown.edu.

The PIR serves as a primary resource for exploration of proteins, allowing users to answer complex biological questions that may typically involve querying multiple sources. In particular, interesting relationships between database objects, such as relationships among protein sequences, families, structures and functions, can be revealed. Functional annotation of proteins requires the association of proteins based on properties beyond sequence homology—proteins sharing common domains connected via related multi-domain proteins (grouped by superfamilies); proteins in the same pathways, networks or complexes; proteins correlated in their expression patterns and proteins correlated in their phylogenetic profiles (with similar evolutionary patterns). The PIR, with its integrated databases and analysis tools constitutes a fundamental bioinformatics resource for biologists who contemplate using bioinformatics as an integral approach to their genomic/proteomic research and scientific inquiries.

Acknowledgments

The PIR is supported by grant P41 LM05978 from the National Library of Medicine, National Institutes of Health. The iProClass and RESID databases are supported by DBI-9974855, DBI-0138188, and DBI-9808414 from the National Science Foundation.

Glossary

Domain An independently folded unit within a protein or a discrete portion of a protein with a unique function. Most proteins are multi-domain. The overall function of a protein is determined by the combination of its domains.

Motif A short conserved region in a protein sequence. Motifs are frequently highly conserved parts of domains.

Protein family classification Classification of proteins into groups with similar structure and/or function based upon comparisons of motifs, domains or family membership rather than pair-wise sequence comparisons.

Protein superfamily or family A set of proteins which share a common evolutionary ancestor as defined by sequence homology. Protein families can be arranged in a hierarchy, in which closely related proteins with a recent common ancestor comprise subfamilies, and larger groups of distant homologs comprise superfamilies.

Suggested Readings

Introduction — Protein Databases

Barker, W. C., Pfeiffer, F., and George, D. G. (1996) Superfamily classification in PIR-International Protein Sequence Database, Methods Enzymol., 266, 59–71.

Bateman, A., Birney, E., Cerruti, L., Durbin, R., Etwiller, L., Eddy, S. R., et al. (2002) The Pfam protein families database, Nucleic Acids Res. 30, 276–280.

Dayhoff, M. O. (1965–1978) Atlas of Protein Sequence and Structure, vol. 1–5, Suppl. 1–3, National Biomedical Research Foundation, Washington, DC.

Falquet, L., Pagni, M., Bucher, P., Hulo, N., Sigrist, C. J., Hofmann, K., and Bairoch, A. (2002) The PROSITE database, its status in 2002, Nucleic Acids Res. 30, 235–238.

Wu, C. H., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z. Z., et al. (2002) The Protein Information Resource: an integrated public resource of functional annotation of proteins, Nucleic Acids Res. 30, 35–37.

Wu, C. H., Xiao, C., Hou, Z., Huang, H., and Barker, W. C. (2001) iProClass: An Integrated, Comprehensive, and Annotated Protein Classification Database, Nucleic Acids Res., 29, 52–54.

PIR-International Protein Sequence Database

Superfamily Classification

Enzyme Nomenclature, Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB) in consultation with the IUPAC-IUBMB Joint Commission on Biochemical Nomenclature (JCBN). (*see* Website: http://www.chem.qmw.ac.uk/iubmb/enzyme/.)

Integrated Protein Classification Database

Garavelli, J. S., Hou, Z., Pattabiraman, N., and Stephens, R. M. (2001) The RESID database of protein structure modifications and the NRL-3D sequence-structure database, Nucleic Acids Res., 29, 199–201.

Huang, H., Xiao, C., and Wu, C. H. (2000) ProClass protein family database, Nucleic Acids Res., 28, 273–276.

PIR Nonredundant Reference Protein Database

Bairoch, A. and Apweiler, R. (2000) The Swiss-Prot protein sequence database and its supplement TrEMBL in 2000, Nucleic Acids Res., 28, 45–48.

National Center for Biotechnology Information (NCBI) taxonomy home page (*see* Website: http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/)

Pruitt, K. D. and Maglott, D. R. (2001) RefSeq and LocusLink: NCBI gene-centered resources, Nucleic Acids Res., 29, 137–140.

Westbrook, J., Feng, Z., Jain, S., Bhat, T. N., Thanki, N., Ravichandran, V., et al. (2002) The Protein Data Bank: unifying the archive. Nucleic Acids Res. 30, 245–248.

PIR System Distribution

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res., 25, 3389–3402.

Eddy, S. R., Mitchison, G., and Durbin, R. (1995) Maximum Discrimination hidden Markov models of sequence consensus, J. Comp. Biol,. 2, 9–23.

McGarvey, P., Huang, H., Barker, W. C., Orcutt, B. C., and Wu, C. H. (2000) The PIR Website: new resource for bioinformatics, Bioinformatics, 16, 290–291.

Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparision, Proc. Natl. Acad. Sci. USA, 85, 2444–2448.

Smith, T. F. and Waterman, M. S. (1981) Comparison of bio-sequences, Adv. Appl. Math., 2, 482–489.

Alignment Algorithms and Prediction

Marcotte, E. M, Pellegrini, M., Thompson, M. J., Yeates, T. O., and Eisenberg, D. (1999) A combined algorithm for genome-wide prediction of protein function, Nature, 402, 83–86.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucleic Acids Res., 22, 4673–4680.

Wu, C. H., Huang, H., and McLarty, J. (1999) Gene family identification network design for protein sequence analysis, Intl. J. Artificial Intelligence Tools (Special Issue Biocomput.) 8, 419–432.

27 Sequence Similarity and Database Searching

David S. Wishart

Introduction

Database searching is perhaps the fastest, cheapest, and most powerful experiment a biologist can perform. No other 10-s test allows a biologist to reveal so much about the function, structure, location or origin of a gene, protein, organelle, or organism. A database search does not consume any reagents or require any specific wet-bench laboratory skills; just about anyone can do it, but the key is to do it correctly. The power of database searching comes from not only the size of today's sequence databases (now containing more than 700,000 annotated gene and protein sequences), but from the ingenuity of certain key algorithms that have been developed to facilitate this very special kind of searching. Given the importance of database searching it is crucial that today's life scientists try to become as familiar as possible with the details of the process. Indeed, the intent of this chapter to provide the reader with some insight and historical background to the methods and algorithms that form the foundation of a few of the most common database searching techniques. There are many strengths, misconceptions and weaknesses to these simple but incredibly useful computer *experiments*.

Similarity Versus Homology

Considerable confusion still exists over the proper use of the terms, *similarity* and *homology*. Therefore, before we can begin any in-depth discussion on sequence similarity and databasae searching, it is important that we make a clear distinction between sequence similarity and sequence homology. Here are two definitions that should help:

- Similarity: In sequence analysis, this refers to the likeness or percent identity between any two sequences. Sequences are similar if they share a statistically significant number of amino acids or nucleotides in approximately the same positions. Similarity does not infer homology, it is only a descriptive term that carries no suggestion of shared ancestry or origin. Similarity is often enumerated by *percent identity* or an *expect (E) value*.
- Homology: In sequence analysis, this refers to a shared ancestry. Two sequences are homologous if they are derived from a common ancestral sequence or if one of the two sequences has diverged (through evolution) to be different in its amino acid or nucleotide sequence from its parent. Homology almost always implies

similarity, but similarity does not always imply homology. Homology is a *qualitative assertion* or *hypothesis*.

As these two definitions suggest, similarity can be quantified while homology has to be justified. Typically, to claim that two sequences are homologous, one usually needs more solid (i.e., experimental) evidence than just a sequence alignment and a high level of shared sequence identity. It is also important to remember that one cannot say two sequences are *X percent homologous* or *Y percent similar*. In fact, the only time a percentage can be given is when one speaks in terms of sequence identity, e.g., these two sequences are Z percent identical. It is OK, however, to say an alignment has a similarity score of X, just as long as one indicates what scoring matrix or scoring protocol was used.

DNA Versus Protein

Prior to performing any database search, a critical decision must be made: Should my query sequence be DNA or protein? The answer to this is clear, but may be surprising to many. Mathematicians have determined that sequence comparison is best carried out when the sequences exhibit *complexity*, meaning that they are composed of a large number of different characters. In this regard, protein sequences with their 20 letter alphabet are far more complex or informationally *richer* than DNA sequences, which only have a 4-letter alphabet. Consequently, database searches using DNA sequences are more likely to yield ambiguous results or potentially false alignments than database searches conducted against protein sequences. It is for this reason that most experienced bioinformaticians insist that DNA sequences be translated to protein sequences before they are submitted for database comparisons. Indeed, the only reasons why one would not want to translate a DNA sequence into the corresponding protein sequence are if the query sequence quality is low (as with ESTs), if the DNA does not code for any protein product (e.g., a DNA regulatory element) or if it corresponds to a tRNA or rRNA gene. *So always translate those DNA sequences!*

Dynamic Programming and Sequence Similarity

Dynamic programming lies at the heart of almost all sequence alignment and database searching routines. Strictly speaking dynamic programming is an efficient mathematical technique that can be used to find optimal *paths* or routes to multiple destinations or in locating paths that could be combined to score some maximum. The application of dynamic programming to sequence alignment was first demonstrated more than 30 years ago by Needleman and Wunsch (1970). The great strength of dynamic programming is that it actually permits a quantitative assessment of similarity, while at the same time showing how two sequences can be globally aligned. The great weakness of dynamic programming is that it is incredibly slow and extraordinarily memory intensive. Strictly speaking, dynamic programming is an N × M algorithm, meaning that the time it takes to execute and the amount of computer memory (RAM) required is dependent on both the length of the query sequence (N) and the cumulative length of the database sequence(s) (M). Given that databases today have an effective length of nearly 250 million amino acids or 4 billion bases, this obviously makes dynamic programming a rather slow approach to database searching. While the dynamic programming algorithm is inherently slow, it is still the most mathematically rigorous method for determining a pairwise sequence alignment. No other technique (not FASTA, nor BLAST) can offer a guaranteed mathematically optimal alignment. While this is a very strong statement, one must always remember that what is mathematically correct may not be biologically correct. In other words, one should always use common sense and a bit of biological intuition when evaluating any given sequence alignment.

Dynamic Programming: The Algorithm

In dynamic programming, pairwise alignments are calculated in two stages. The first stage is called the *scoring stage*. It involves putting the two sequences of interest (e.g., sequence A and sequence B) on the two axes of a table and then progressively filling that table from the upper left to the lower right with integer scores related to the pairwise similarity of each base or amino acid in the two sequences. These scores are initially obtained by comparing each letter in sequence A to each letter in sequence B using a scoring matrix such as the BLOSUM62 matrix. These similarity scores are then further modified using a special recursive scoring function shown in Equation 1.

$$S_{ij} = s_{ij} + \mathrm{Max}\left\{\begin{array}{ll} s_{i-l,j-l} & \text{or} \\ \mathrm{Max}\ s_{i-x,j-l} + w_{x-l}\ (2 < x < i) & \text{or} \\ \mathrm{Max}\ s_{i-l,j-y} + w_{y-l}\ (y < 2 < i) & \end{array}\right\} \qquad \text{[Eq. 1]}$$

Where S_{i-j} is the *net* score for the alignment ending at i in sequence A and j in sequence B, s_{ij} is the score for matching i with j, w_x is the score for making a x long gap in sequence A, and w_y is the score for making a y long gap in sequence B. Figure 1 illustrates how this table can be filled out using a simple scoring procedure where identical amino acids are given a score (s_{ij}) of 1 for a match, 0 for a mismatch and no gap penalty. In this figure we are attempting to align the sequence *GENES* (sequence A) with the sequence *GEMS* (sequence B).

Once this table is filled out, the second (*traceback*) stage is undertaken wherein one scans through the matrix in a diagonal fashion from the lower right to upper left to look for the highest scores entered in the matrix. The path that is chosen is actually a series of *maximum* numbers. When all the scores in this optimal path are added together, it gives a quantitative measure of the pairwise sequence similarity while at the same time defining which letters in sequence A should be matched with the letters in sequence B. The traceback route(s) along with the two possible alignments for the GENES vs GEMS example are shown in the lower part of Fig. 1. In this case the similarity score for both alignments was 3 + 2 + 2 + 1 = 8.

The original Needleman-Wunsch (NW) algorithm was specifically developed to perform global pairwise sequence alignments. In other words, the scoring function and traceback procedure implicitly assumes that the optimal alignment will be found by going from the lower right corner to the upper left corner of the matrix. A simple modification to the NW algorithm introduced by Smith and Waterman (1981) showed that dynamic programming could actually be adapted to perform local alignments. In other words, the path for the traceback procedure did not have to start and end at the edges of the search matrix, but it could start and end internally (i.e., within the search matrix). Such an alignment, once found, would be locally optimal. The advantage of identifying a locally optimal alignment, particularly for proteins, is that it may be possible to pull out small functionally related domains that would normally be missed by a global alignment algorithm. Because of their potential advantage in identifying remote or frequently missed sequence similarities, local alignment techniques have been the subject of considerable interest development over the past 15 years. Two fast local alignment methods in particular, namely FASTA and BLAST, arose from this work. These heuristic database search approaches will be discussed in more detail later on in this chapter.

Scoring Matrices

The scoring system illustrated in Fig. 1 used a simple match/mismatch scoring function. Such a scoring system can be described by a simple table where we place all 4 bases or all 20 amino acids on both the X and Y axes and indicate with a 1 or a 0 (inside the table) whether any two bases or any two amino acids match or do not match. This type of scoring table is commonly referred to as an *identity* or *unitary* scoring matrix and it is still the most commonly used scoring system for aligning or matching nucleic acid sequences. Scoring matrices are key to the success of dynamic programming. They are also its Achilles heel. Simply stated, scoring matrices try to capture billions of years of evolutionary information and tens of thousands of complex molecular interactions and reduce it all to a table of 200 integers. Perhaps this is scientific reductionism at its most extreme. Nevertheless, it seems to work (most of the time). Over the past 25 years more than a dozen amino acid scoring matrices have been described in the literature, however there are only two types of scoring matrices that seem to have survived the test of time: 1) PAM scoring matrices; and 2) BLOSUM scoring matrices. These scoring schemes are described in the following.

The Dayhoff (PAM) Scoring Matrices

Until recently the most common scoring matrices for protein sequence comparisons were based on the mutation data matrix (MDM_{78}) developed by Margaret Dayhoff and coworkers in 1978. These were derived using the Point Accepted Mutation (PAM) model of evolution originally proposed by Dayhoff. Using a set of proteins that were >85% identical, Dayhoff and her colleagues manually aligned these proteins, taking great pains to ensure the alignments were completely unambiguous. From these groups of alignments, an amino acid similarity ratio (R_{ij}) was calculated for each pair of amino acids as shown in Equation 2:

$$R_{ij} = q_{ij} / p_i p_j \qquad \text{[Eq. 2]}$$

Where q_{ij} is the observed frequency that residues i and j replaced each other in the set of aligned proteins and p_i and p_j are the observed frequencies of amino acids i and j in the complete set of proteins. Therefore, for conservative replacements (such as Y for F), R_{ij} would be expected to be greater than 1, while for unfavorable replacements, R_{ij} would be less than 1.

The initial PAM scoring matrix was calculated such that the probabilities in the scoring table would represent the average mutational change when one residue in 100 mutates (1% point accepted mutations or 1 PAM). This particular matrix was called the PAM-1 matrix. By assuming that the mutations that occur in a protein sequence are essentially not correlated with previous mutations, it is possible to use the mathematics of Hidden Markov Models to calculate a mutational probability matrix that has undergone N percent PAM's. To do so, the PAM-1 matrix is multipled by itself N times.

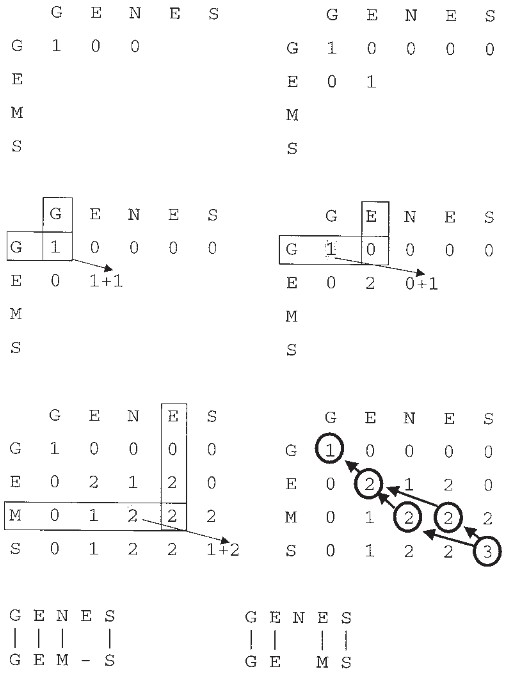


Fig. 1. A simple illustration of how dynamic programming is performed in the alignment of two short peptide sequences (sequence A = GEMS; sequence B = GENES). The upper tables illustrate the progressive filling of the scoring matrix using the recursion relationship in Eq. 1 and the following rules: a match =1, a mismatch = 0, and a gap insertion = 0 (i.e., w = 0). The lower right table illustrates the traceback procedure. The two possible alignments between GEMS and GENES, determined via dynamic programming, are shown at the bottom.

Using this simplified model of evolution Dayhoff created an entire family of scoring matrices, with the most useful being the PAM-120 and PAM-250 matrices. These matrices were created by multiplying the PAM-1 matrix by itself 120 and 250 times, respectively. As a general rule, to align sequences that are highly divergent, the best results are obtained with matrices having higher PAM values. PAM-120 and PAM-250 can be viewed as being the typical scoring matrices of proteins that have been separated by 120 million (PAM-120) and 250 million (PAM-250) years of evolution. Dayhoff also converted the PAM-120 and PAM-250 matrices into log-odds matrices. To do so, the natural log of these R_{ij} ratios were determined and the new numbers (S_{ij}) were substituted into the scoring matrix. The log-odds form of the PAM-250 matrix is called MDM$_{78}$ and this is the one that Dayhoff recommended for use in general sequence comparisons. The MDM$_{78}$ or PAM-250 matrix is shown in Fig. 2. As can be seen in this matrix, large positive values indicate a high level of conservation or chemical similarity between a pair of amino acids. Large negative values indicate very little chemical similarity and represent nonconservative mutations. In principle the sum of these log-odds scores over the length of any pair of aligned sequences is equal to the natural log of the probability that the two sequences are related.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	4															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-4	-4	-4	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	-1	-1	-3	0	-1	-1	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	3				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-2	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Fig. 2. The PAM-250 (or MDM_{78}) scoring matrix for amino acid substitutions.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4																			
R	-1	5																		
N	-2	0	6																	
D	-2	-2	1	6																
C	0	-3	-3	-3	9															
Q	-1	1	0	0	-3	5														
E	-1	0	0	2	-4	2	5													
G	0	-2	0	-1	-3	-2	-2	6												
H	-2	0	1	-1	-3	0	0	-2	8											
I	-1	-1	-3	-3	-1	-3	-3	-4	-3	4										
L	-1	-1	-3	-4	-1	-2	-3	-4	-3	2	4									
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Fig. 3. The BLOSUM62 scoring matrix for amino acid substitutions.

The BLOSUM Scoring Matrices

A more recent addition and increasingly popular set of scoring matrices are the Block Substitution Matrix (BLOSUM) scoring matrices. A key difference between the BLOSUM and PAM matrices is that the BLOSUM matrices were constructed without an explicit model of evolution. In particular, the substitution probabilities or target frequencies were estimated from the BLOCKS database, which is a database of local (contiguous) multiple alignments for various distantly related sequence blocks or domains found in proteins. As with the PAM matrices there is a numbered series of BLOSUM matrices, such as BLOSUM30, BLOSUM62, and BLOSUM90, however the numbering scheme is essentially in reverse order relative to the PAM matrices. Specifically, the BLOSUM numbers refer to the maximum level of pairwise sequence identity in the selected sequence BLOCKS used to prepare the scoring matrix. The substitution frequencies for the BLOSUM62 matrix are influenced primarily by those sequences sharing less than 62% sequence identity. In general one would use the BLOSUM90 matrix to compare very similar sequences and the BLOSUM30 matrix to compare very different sequences or to detect more remote sequence similarities. The BLOSUM62 scoring matrix, which is the default scoring matrix used in BLAST, is shown in Fig. 3.

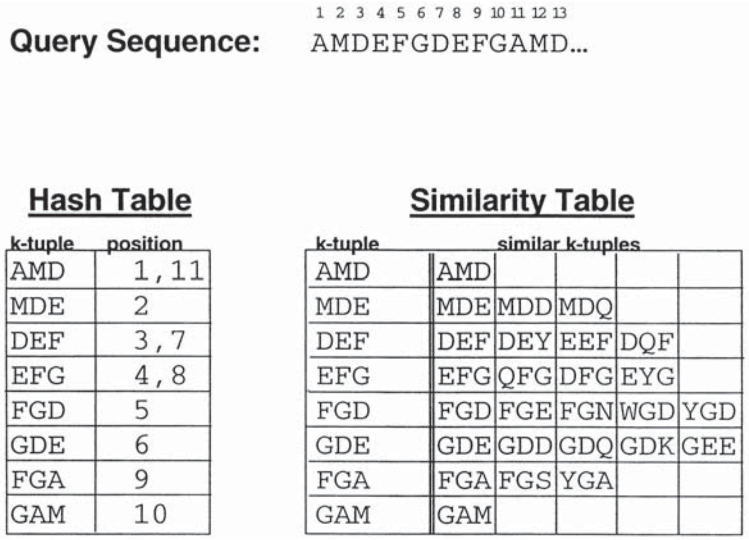
Gap and Gap Extension Penalties

In addition to scoring matrices, the use of gap and gap extension penalties can also increase the sensitivity of a sequence search and the utility of the resulting alignment. Gaps and gap extensions must often be introduced in alignments to accommodate deletions or insertions that may have occurred during the evolutionary divergence of one sequence from another.

Dynamic programming algorithms can be easily modified during the traceback step to include gap penalties and gap extension penalties. Most commonly, a fixed deduction is applied to the alignment score when a gap is first introduced then an extra deduction is added which is proportional to the length of the gap. Typically the gap opening penalty is called G and the gap extension penalty is called L. Therefore the total deduction for a gap of length n would be $G + Ln$. The selection of gap parameters is very empirical and this represents one of the greatest weaknesses in dynamic programming and in sequence alignment in general. For the BLOSUM62 matrix, it is usually recommended that a high gap penalty (G) of 10 to 15 be used along with a low value for L (1 or 2). As a rule of thumb, the gap penalty G should be approximately equal to the difference between the highest and the lowest number in the chosen scoring matrix.

Fast Local Alignment Methods

Dynamic programming is a superb method for comparing two sequences and obtaining a global (i.e., entire) sequence alignment. However, if one wishes to perform 700,000+ comparisons using this approach—typical of a database search—it could easily take several hours on a very fast computer. Given that most of us do not want to wait that long for an answer, there has been a great deal of effort over the past 15 years directed at developing methods to improve search speeds so that database searches could be done more quickly. However, improvements in speed usually come with a sacrifice in accuracy or sensitivity. Nevertheless, the advent of such



1 2 3 4 5 6 7 8 9 10 11 12 13

Query Sequence: AMDEFGDEFGAMD...

Hash Table

k-tuple	position
AMD	1,11
MDE	2
DEF	3,7
EFG	4,8
FGD	5
GDE	6
FGA	9
GAM	10

Similarity Table

k-tuple	similar k-tuples				
AMD	AMD				
MDE	MDE	MDD	MDQ		
DEF	DEF	DEY	EEF	DQF	
EFG	EFG	QFG	DFG	EYG	
FGD	FGD	FGE	FGN	WGD	YGD
GDE	GDE	GDD	GDQ	GDK	GEE
FGA	FGA	FGS	YGA		
GAM	GAM				

Fig. 4. An example of a hash table (left) and a similarity table (right) prepared for a short peptide sequence (top). The numbering associated with each triplet (a word of length 3) is based on the position of the first residue. The sets of similar triplets were identified using similarity scores derived from the BLOSUM62 matrix. The cutoff score was 13.

fast, sensitive heuristic search algorithms such as FASTA and BLAST has revolutionized the process and frequency of sequence comparison and database searching. Their tremendous speed advantages seem to far outweigh their potential shortcomings in sensitivity.

Most fast local alignment algorithms are based on breaking the query sequence down into small overlapping subsequences, i.e., words or k-tuples (a k-tuple is a word or character string that is k letters long), and placing their relative sequence locations into what are called *hash tables*. Hash tables are lists of locations or addresses that allow computers to wade through and compare data more efficiently. Once this hash table is built, it is possible to compare other database sequences to the query sequence very quickly. In fact, it turns out that these hash-based methods scale linearly with the size of the database (i.e., they are M-type algorithms as opposed to N × M algorithms). This kind of algorithmic scaling makes them excellent choices to scan very large (and rapidly growing) databases with little cost in time. Figure 4 illustrates how a query sequence could be broken down into a hash table and how each k-tuple could be assigned to a group of similar k-tuples. This is typically the first step performed in a BLAST search.

BLAST

Basic Local Alignment Search Tool (BLAST) is the most commonly used sequence database search tool in the world. Since its introduction as a web-based service in 1995, BLAST has become *the standard* for sequence alignment and database searching. Key to BLAST's success, both as an algorithm and as a program, has been its use of statistical methods to simultaneously accelerate its search speed and to increase its

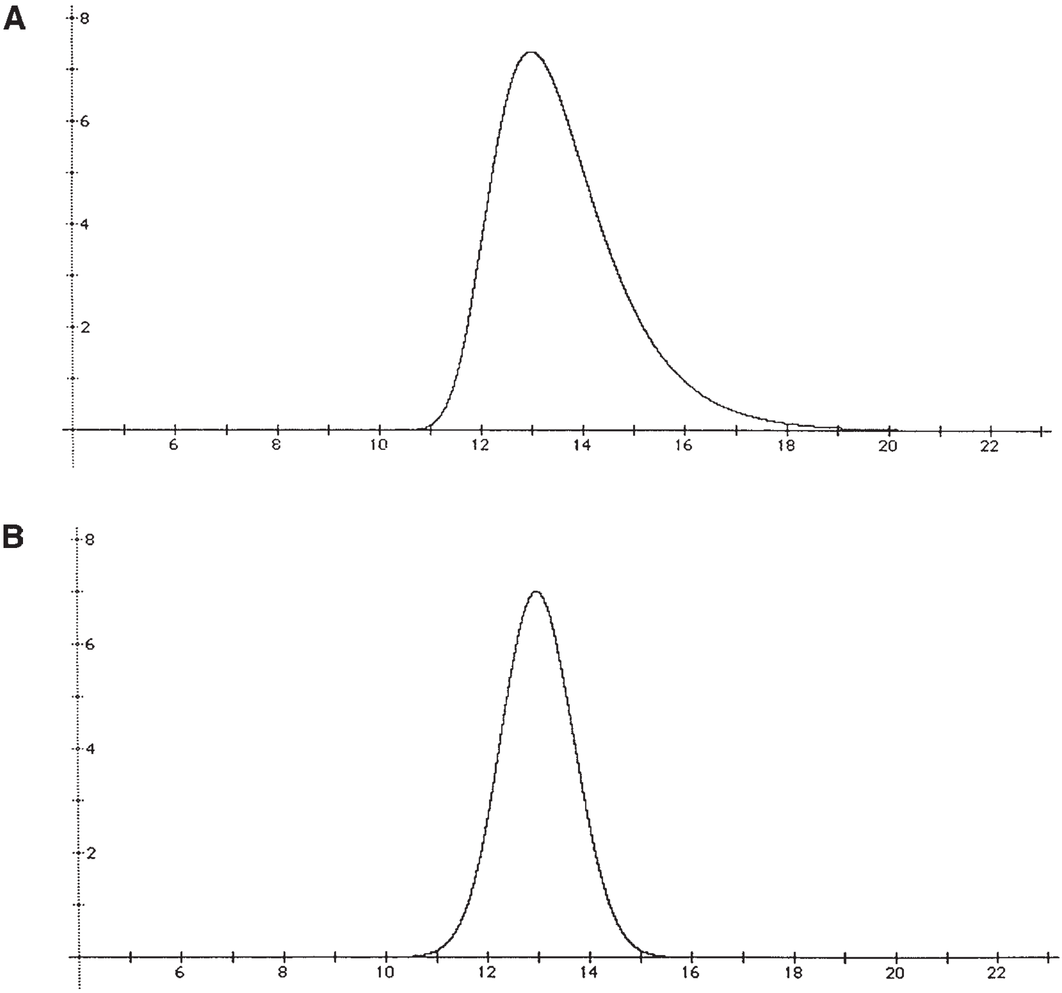


Fig. 5. A comparison between the shapes of an Extreme Value Distribution **(A)** and a Normal or Gaussian distribution **(B)**. The mean (μ) for both distributions is 13.

search sensitivity. In the late 1980s and early 1990s, Altschul and Karlin (two statisticians at Stanford) showed that the significance of pair wise sequence alignments could be rapidly determined or in essence, *predicted* using what is called an *Extreme Value distribution* (*see* Equation 3).

$$P(x) = e^{-x}e^{-e^{-x}} \qquad \text{[Eq. 3]}$$

Where P(x) is the probability of some event getting a score of x. This particular distribution appears reasonable at modeling the distribution of scores or events that are quite rare and which are slightly dependent (as opposed to independent) of each other. Consequently, the Extreme Value distribution differs from a Normal or Gaussian distribution by having a long tail (called a *right-handed skewness*) as shown in Fig. 5. This means that an alignment that scores high and is considered significant for a

Gaussian distribution would likely be considered insignificant in an Extreme Value distribution. By using this special kind of statistical distribution and the probabilities that could be calculated from it, Altschul and coworkers demonstrated that it was possible to rapidly identify, assess, and extend locally similar sequence blocks more quickly and more accurately than other competing alignment programs.

Simply stated, BLAST takes a query sequence and searches a given sequence database for high scoring k-tuples (words) or high scoring segment pairs (HSPs) that exceed a certain scoring threshold (T). This process is illustrated schematically in Fig. 6. Note that unlike earlier database searching algorithms, BLAST does not require HSPs to match exactly. Instead it refers to a precalculated look-up table of highly similar HSPs. This allows BLAST to use a longer word length (3 letters for proteins, 11 letters for DNA) without compromising sensitivity. After the HSPs are identified, BLAST attempts to identify locally dense regions of HSPs and then tries to extend them in either direction until the cumulative alignment score starts dropping (*see* Fig. 7). During the HSP identification and extension phases, BLAST uses rigorous statistical measures to assess significance. This improves on sensitivity and saves on time because the HSP search algorithm does not need to evaluate and extend false leads. In addition to these statistical *speed-ups* BLAST also takes advantage of several advanced programming techniques such as preloading the sequence database into RAM, incorporating finite state transition methods, and exploiting the large *word* size in modern computers to read and process data more quickly.

Historically, one of the key limitations of BLAST (version 1.4 and earlier) was the fact that its sequence alignments had to be broken down into collections of ungapped local alignments or sequence blocks. Thus the term *Local Alignment Search* was born. The difficulty in converting local alignments to more global alignments (which one typically gets with dynamic programming methods or FASTA searches) was overcome with the introduction of Gapped BLAST in 1997. This is now the default BLAST service offered by the NCBI and its mirror sites.

Gapped BLAST offers two key improvements over earlier versions of BLAST. First, an improvement in search speed was attained by requiring that at least two words had to occur in a given window length (A = 40 residues) in order for a given subsegment to be considered significant. This allows the program to ignore many other random HSP hits and to concentrate on the ones most likely to lead to an extended HSP. The second and most important improvement, however, lies in the fact that BLAST now handles gaps explicitly. The strategy adopted in gapped BLAST is to initially identify a HSP in the middle of the highest scoring window of 11 residues. Using this maximal segment pair, BLAST uses the Smith-Waterman algorithm to extend this alignment in both directions until it falls below a fixed percentage of the previous top score. This process is analogous to the way ungapped BLAST assembles HSPs. Another recent improvement in BLAST has been the introduction of composition statistics. This permits the E (expect) values to account for the differing amino acid or base composition of the query sequences and database hits. This little innovation effectively allows one to apply a *custom* scoring procedure for each sequence in the database. The net result is that the inclusion of composition statistics improves the sensitivity and specificity of the database search by reducing the number of false positives.

BLAST is under constant development and improvements to both the algorithm and its associated databases are continuing. A periodic check of the NCBI website will allow any user to stay current with the latest news on BLAST.

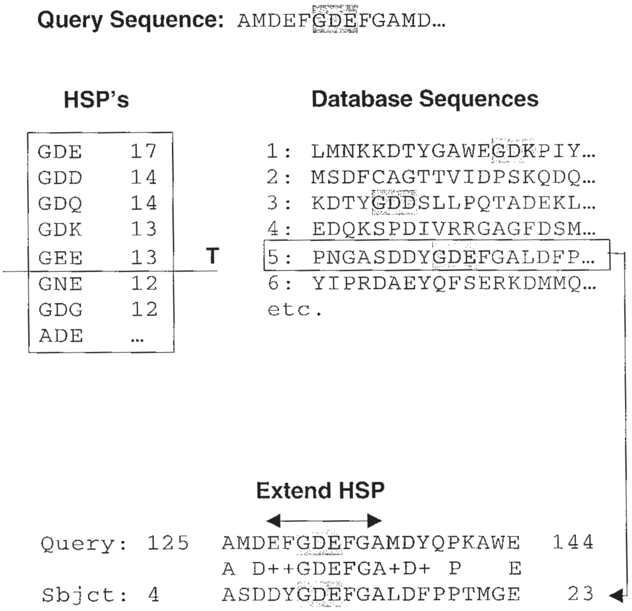


Fig. 6. A schematic illustration of how BLAST identifies and extends HSPs. The HSP in this case is centered around the triplet *GDE*. The Threshold (T) cutoff score in this case is 13, so only those triplets in the database set (right) with scores > 12 will be identified as HSPs. In the lower panel a sequence (sequence 5) has been chosen for further analysis and the initial HSP has been extended in both directions, producing the alignment seen here.

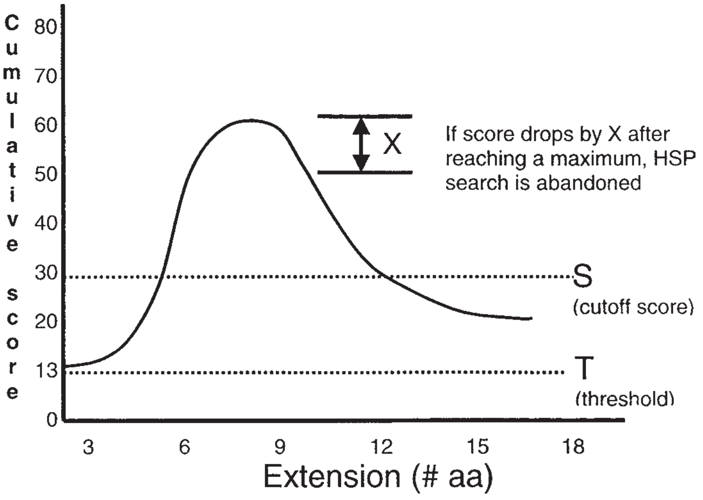


Fig. 7. During the HSP extension phase, a cumulative score is kept between the query and database sequence as the HSP is lengthened. If the HSP is of high quality the score will rapidly climb. Eventually the cumulative HSP score will reach a maximum as the similarity weakens at one end of the sequence or the other. BLAST stops the HSP extension once the cumulative HSP score drops by more than X (a drop-off parameter defined in BLAST).

Running BLAST (The Web Version)

BLAST can be freely and easily accessed from any number of websites from around the world. Some of these include the NCBI (*see* Website: http://www.ncbi.nlm.nih.gov/BLAST), the EBI (*see* Website: http://www.ebi.ad.uk/blastall/), or the Canadian Bioinformatics Resource (*see* Website: http://www.cbr.nrc.ca/blast). BLAST can also be downloaded and run locally or it can be accessed through a variety of commercial programs (most of which have customized front-ends to access the NCBI or EBI sites). For the purposes of this chapter, we will focus on the version of BLAST offered through the NCBI website.

Different Flavors of BLAST

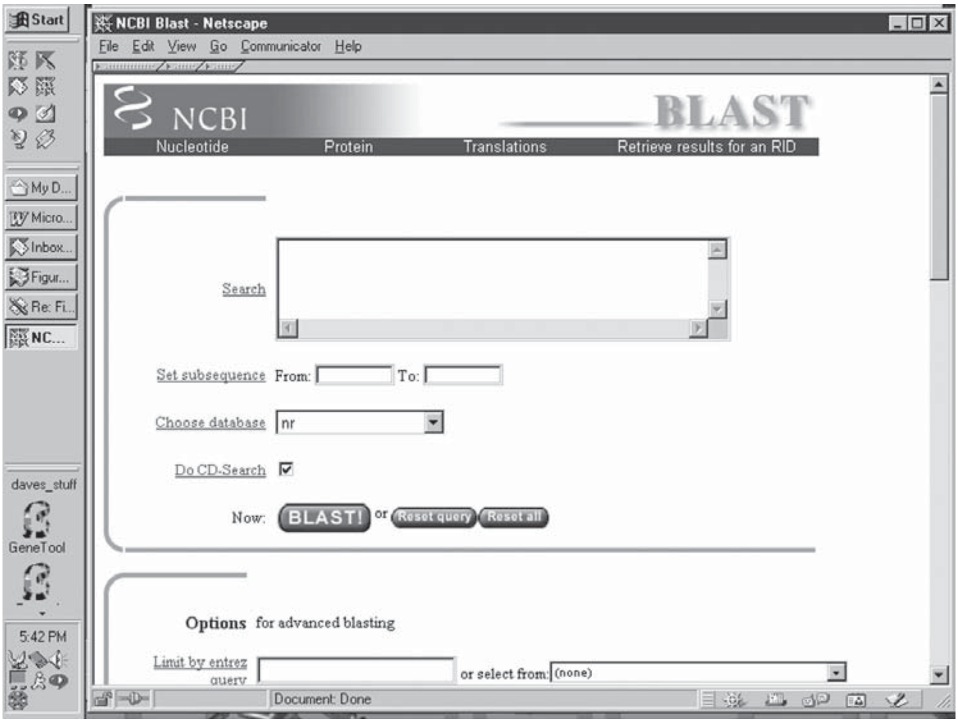
When first logging onto the NCBI site, you will notice a rather large number of BLAST options. These represent the different *flavors* or variations of BLAST that have been developed to accommodate the different needs of users and the different settings required for different types of searches. Currently the NCBI offers at least eight different types of BLAST database searches:

1. BLASTP: Searches user-selected protein databases with protein sequence queries.
2. BLASTN: Searches user-selected nucleic acid databases with DNA/RNA queries.
3. BLASTX: Searches 6-frame translated DNA databases with protein queries.
4. TBLASTN: Searches protein databases with 6-frame translated DNA queries.
5. TBLASTX: Searches translated DNA databases with translated DNA queries.
6. PSI-BLAST: Searches user-selected protein databases with protein sequence profiles.
7. PHI-BLAST: Searches user-selected protein databases for protein sequence patterns.
8. MEGABLAST: Searches entire genome against user-selected nucleic acid databases.

BLASTP is useful for looking for moderate-to-distant relationships among protein sequences. BLASTN is primarily for identifying relatively strong matches between long DNA sequences. BLASTX is most commonly used for analyzing new DNA sequences and EST's (expressed sequence tags). TBLASTN is useful for finding undetected or unexpected protein coding regions (genes) in newly sequenced DNA. TBLASTX is often used in EST analysis. PSI-BLAST is particularly useful for identifying extremely distant or difficult-to-identify relationships between protein sequences. Indeed, with the introduction of PSI-BLAST, BLASTP has largely become obsolete. PHI-BLAST is used to look for user-specified protein sequence patterns. MEGABLAST is best used for whole genome comparisons at the nucleotide level. It is optimized for aligning sequences that differ only slightly and is up to 10 times faster than BLASTN.

BLAST Input

Once you have chosen your desired flavor of BLAST you are ready to submit your sequence request(s). The standard BLAST input window is illustrated in Fig. 8. In most cases you only need to paste in your sequence, choose your database and press the **Submit** button. A request ID (RID) number will be immediately assigned to you and a few seconds later you can press the **Format** button to obtain your search results. BLAST has been carefully designed so that its default values will generally provide a pleasing first-pass result. However, to properly use BLAST and to properly interpret its output, it is important to understand something about its input options. Here is a brief description of some of the key terms or parameters.



Search

Set subsequence From: To:

Choose database nr

Do CD-Search ☑

Now: BLAST! or Reset query Reset all

Options for advanced blasting

Limit by entrez query or select from: (none)

Document: Done

Fig. 8. A screen shot of the NCBI BLAST server. A description of the options and functions is given in the text.

The Sequence Window

This box allows users to type or paste a query sequence for searching. The sequence should be in FASTA format, which means that a > character must appear on one line (the sequence name is optional) and the sequence should appear on the other lines. Users may also type in a gene index (GI) or Genbank accession number, instead of a sequence. This latter option can save quite a bit of time.

Database Selection

NCBI and its mirror sites offer a wide range of both general and specialized databases of widely varying scope and quality. There are 8 different protein databases and 14 different DNA databases, both of which include several organism-specific databases (*E. coli*, yeast, drosophila), a patented sequence database (pat), and monthly updates (month). Outside of interchanging BLASTN with BLASTP, the most common error committed by BLAST users is in (inadvertently) choosing the wrong database. The NR (Nonredundant) database contains the most sequence data and is the default, however, it is not necessarily nonredundant and it certainly does not offer the most complete annotation (especially for proteins). In this regard, the SWISS-PROT database, which is a secondary or curated database, is the most fully annotated and perhaps the least redundant protein sequence database. The GenBank nr database is still the best database for genomic data although it, too, has its share of errors or omissions. The Sequence Tagged Site (STS), Expressed Sequence Tagged (EST), High Throughput Genomic Sequence (HTGS), and Genomic Survey Sequence (GSS) databases are notorious for their high frequency of sequence errors.

CD Search

The Conserved Domain (CD) search is a relatively new option that is particularly helpful for protein sequence searches. The CD search scans the Conserved Domain Database, which is a collection of preidentified functional or structural domains derived from the Pfam and Smart databases. It uses Reverse Position Specific BLAST to enhance the sensitivity of the search. When turned on, the CD search option can reveal many hidden domains and offer colorful structural insights through the Cn3D molecular viewer (*see* Chapter 32).

Filters

One can still encounter problems with sequence complexity, regardless of whether one chooses to work exclusively with proteins or not. This occurs when either the query sequence or the database sequence contains regions of biased sequence composition or low complexity regions (LCRs). Examples of LCRs are repeated stretches of the same amino acid or the same base (homopolymeric runs), short period repeats (such as CA repeats in DNA), and large repetitive elements (such as Alu repeats, LINEs and MERs). Wooton and Federhen (1996) have developed several approaches to identify these LCRs and to screen them out from database searches. The SEG program specifically looks at proteins and screens out their LCRs by replacing the amino acids in the suspect region with a series of Xs. The DUST program does the same for nucleic acids. In proteins, LCRs most likely exist as nonglobular regions and likely arise from errors in DNA replication (polymerase slippage) or unequal crossing over. The NCBI BLAST server and most of its mirror sites offer LCR screening as a default option. This option should always be kept on unless one is dealing with a special situation or an unusual query sequence.

Expect

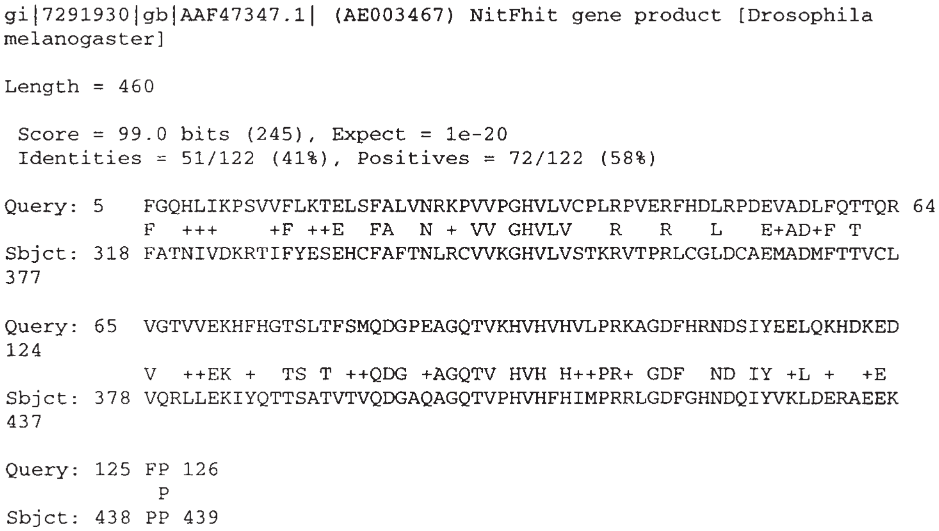
This refers to the expected number of chance HSP alignments with scores equal to or greater than S (*see* following section "S - Alignment Score"). A very small value of E (something close to 0, say 2.1e-73) is characteristic of a highly significant match. A high E value (say greater than 1) is characteristic of an insignificant match. The default Expect value in BLAST is 10. For most sequences one should never trust an alignment that has an E value of more than 0.01

Word size

This is the length of the word or k-tuple that is scanned. Larger values for W (11–12) are typical for DNA searches. Smaller values (3–4) are typical for protein searches. The longer the word size, the faster the search. However, longer word sizes also reduce the sensitivity.

S - Alignment Score (Cutoff)

This is the score determined through comparing sequence matches of High Scoring Segment Pairs (HSPs) to the scoring matrix. S is also called the HSP cutoff score. For an alignment to be reported by BLAST it has to have an HSP with a score greater than or equal to S.



```
gi|7291930|gb|AAF47347.1| (AE003467) NitFhit gene product [Drosophila
melanogaster]

Length = 460

 Score = 99.0 bits (245), Expect = 1e-20
 Identities = 51/122 (41%), Positives = 72/122 (58%)

Query: 5   FGQHLIKPSVVFLKTELSFALVNRKPVVPGHVLVCPLRPVERFHDLRPDEVADLFQTTQR 64
            F  +++    +F ++E  FA  N + VV GHVLV    R    R    L   E+AD+F T
Sbjct: 318 FATNIVDKRTIFYESEHCFAFTNLRCVVKGHVLVSTKRVTPRLCGLDCAEMADMFTTVCL
377

Query: 65  VGTVVEKHFHGTSLTFSMQDGPEAGQTVKHVHVHVLPRKAGDFHRNDSIYEELQKHDKED
124
            V  ++EK +  TS T ++QDG +AGQTV HVH H++PR+ GDF  ND IY +L +  +E
Sbjct: 378 VQRLLEKIYQTTSATVTVQDGAQAGQTVPHVHFHIMPRRLGDFGHNDQIYVKLDERAEEK
437

Query: 125 FP 126
             P
Sbjct: 438 PP 439
```

Fig. 9. BLASTP output from a database search using Human Fragile Histidine Triad Protein (P49789) as the query sequence.

T - Threshold

This is the threshold score that a short word (HSP) or k-tuple must have in order for BLAST to consider it significant. T is a function of the scoring matrix used in the program. This is now automatically determined when the user selects their preferred scoring matrix.

Matrix and Gap Costs

These were detailed in the previous sections of this chapter.

BLAST Output and Assessing Sequence Similarity

An example of the alignment output of a typical BLAST run is shown in Fig. 9. BLAST also provides alignment lists and image-mapped *domain* graphs to facilitate querying or analyzing the results (not shown here). In addition to the name, accession number, and length of the protein, the alignment produced by BLAST includes data on the alignment score, the *Expect* value and other measures. The Bit Score (the first number) is a normalized score (S) derived from the raw score (given in parentheses). The raw score refers to the summed HSP scores (*see* S above) and is of course dependent on the scoring matrix used. The bit score is independent of the scoring matrix chosen, so one may use the bit score as a means of consistently comparing alignments obtained with different scoring matrices (e.g., BLOSUM62 vs BLOSUM30). While the bit score is generally quite useful, the key parameter to evaluating any BLAST output is the *Expect* value. This refers to the expected number of *chance* HSP alignments with scores equal to or greater than the bit score based on the size of the database, the length of the query sequence, and the scoring matrix used. Ideally this should be less than 1; otherwise this suggests that the alignment you have found could be just a spurious match. As a rule, one should not trust alignments with an Expect value

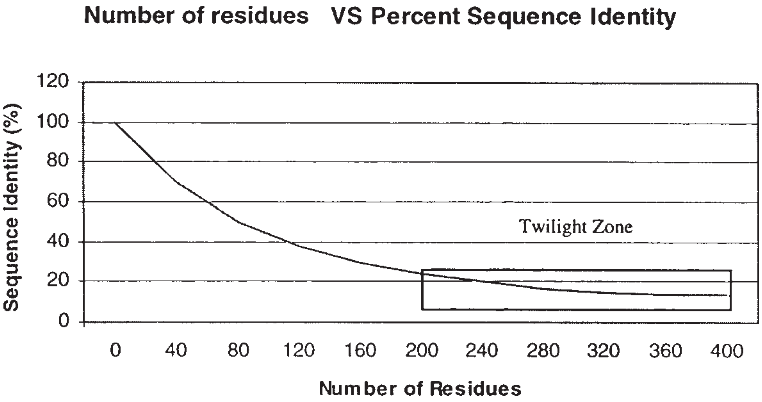


Fig. 10. A plot showing the relationship between sequence identity and sequence length for proteins. Sequences falling above this line are considered to be related (i.e., homologous), while sequences falling below this line are likely unrelated.

greater than 0.01. However, as with anything in sequence alignment, one should assess the alignment by taking into consideration the type and location of matching residues, the number of gaps and the biological or functional relationship between the query and matching sequence before making a decision.

In addition to the bit scores and Expect values, BLAST provides information on the number and percentage of matching residues (Identities) as well as the number and percentage of similar residues (Positives) and Gaps. These are very useful and can be critical to assessing the presumed relationship between two sequences. As a general rule, if two sequences are longer than 100 residues and are more than 25% identical after suitable gapping, it is very likely that they are related. If two sequences are 15–25% identical they may be related but it is usually necessary to perform further tests to confirm their putative relationship. If two sequences are less than 15% identical, they are probably not related. These rules (courtesy of R.F. Doolittle) are summarized in a graph shown in Fig. 10.

Most BLAST searches result in at least one or more significant matches. The determination of whether a match is significant is dependent not only on the numbers (Expect, bit score, %ID) that you obtain, but also on the type of relationship you are trying to identify. For example, if one is interested in finding out whether a query protein is structurally similar to something already in the database, then a match with very low sequence identity (<20%) or a modest Expect value (0.5) could very well exhibit a similar three-dimensional fold. However if one is trying to ascertain whether two proteins have the same function, one typically requires a much higher level of sequence identity (e.g., >50%) or a much lower Expect value. This simply reflects the fact that structure is generally more conserved than either function or sequence. With the rapid growth in whole genome sequence data, there is an unfortunate trend to use large-scale sequence alignments to ascertain the function or putative functions of thousands of proteins using very generous (i.e., low sequence identity) criteria. The result is that many proteins are being functionally misassigned (*see* Chapter 26). While sequence alignment is useful, it is important that alignment data be combined with experimental (structural, enzymatic, evolutionary) data to get the most complete and accurate picture possible.

Recommendations

If one is working with protein sequences or coding segments, the best approach to analyze a query sequence is by using PSI-BLAST. As a general rule, it is best to work with proteins greater than 50 residues, as smaller sequences can lead to spurious results. The default parameters for PSI-BLAST are usually sufficient for an initial search. In using PSI-BLAST it is important to remember to press the **Iterate** button. PSI-BLAST is an iterative process wherein each iteration allows the program to *learn* about the significant features of any previously identified set of matching sequences. Often one will have to press the **Iterate** button 3–4 times before the search converges. If no obvious homologs are apparent after a few iterations, try changing the scoring matrix to BLOSUM30 or PAM250. Once a set of proteins has been identified from this search (using the criteria discussed earlier), one should then ascertain whether the protein exhibits domain-like features. These features may be identified by looking for contiguous segments of the query protein (50–150 residues) matching distinctly different types or classes of proteins in the database. This is best observed through a visual inspection of the colored alignment graph at the top of the BLAST output window. It can also be seen using the CD search option. Once one has identified the different domains in a query protein, it is often a good idea to break the protein sequence into its constituent domains and to perform a PSI-BLAST search with each of these domains against the Protein Data Bank (PDB). This allows one to investigate whether the query protein may have some structural homologs. An equally good idea is to search the SWISS-PROT database (*see* Website: http://www.expasy.ch) as a means of retrieving additional annotation or information about the protein of interest. Further analysis, e.g., using multiple alignments, hydrophobicity analysis, signal site or PROSITE predictions, secondary structure prediction, or Medline searches. should always be carried out to help confirm or further identify important features of any newly sequenced protein.

However, bioinformatics and computer-based analysis can only take you so far. Indeed, database searching is essentially a computer experiment that helps generate (and sometimes confirm) a hypothesis. As with any experiment or hypothesis, it must be confirmed, controlled, and verified. If one neglects to use biological intuition or ignores the results of wet-bench experiments, or if one does not attempt to confirm the results with another laboratory measurement, then the results of a database search probably have little meaning or merit. Despite these cautions and caveats, it is perhaps safe to say that database searching is probably the fastest, easiest and most revealing experiment a biologist can perform.

Glossary and Abbreviations

Algorithm a defined sequence of actions or computational steps that perform a specific task.

Alignment the process of shifting letters in two character strings to create or highlight character matches between the two character strings.

BLAST a very fast and very powerful heuristic sequence alignment and database searching algorithm. BLAST uses well developed statistical methods to assess and identify significant matches and alignments.

Dynamic Programming an efficient mathematical technique that can be used to find optimal "paths" or routes to multiple destinations. Dynamic programming belongs to a special class of optimization or minimization techniques.

FASTA one of the first fast heuristic sequence alignment algorithms to be developed. FASTA made sequence searching through large sequence databases feasible.

Gap Penalty the numerical cost of opening or adding a gap to a sequence alignment. Gap penalties typically have values similar to those in a scoring matrix.

Heuristic an economical or intuitive strategy for determining a solution to a problem to which an exact solution is computationally impossible or intractable.

Homology in sequence analysis, this refers to a shared ancestry. Two sequences are homologous if they are derived from a common ancestral sequence or if one of the two sequences has diverged (through evolution) to be different in its amino acid sequence from its parent.

Scoring Matrix a table of numbers that provides the "cost" of substituting one amino acid or one base for another. Dynamic programming algorithms use scoring matrices to determine optimal sequence alignments.

Similarity in sequence analysis, this refers to the likeness or percent identity between any two sequences. Sequences are similar if they share a statistically significant number of amino acids in approximately the same positions.

Suggested Readings

DNA Versus Protein

Baxevanis, A. D. and Ouellette, B. F. F. (2001) Bioinformatics A Practical Guide to the Analysis of Genes and Proteins 2nd Edition, John Wiley & Sons, NY.

Doolittle, R. F. (1986) Of URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences, University Science Books, Mill Valley, CA.

Dynamic Programming and Sequence Similarity

Needleman, S. B. and Wunsch, C. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48, 443–453.

Dynamic Programming: The Algorithm

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool, J. Mol. Biol. 215, 403–410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25, 3389–3402.

Lipman, D. J. and Pearson, W. R. (1985) Rapid and sensitive protein similarity searches, Science 227, 1435–1441.

Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison, Proc. Natl. Acad. Sci. USA 85, 2444–2448.

Pearson, W. R. (2000) Flexible sequence similarity searching with the FASTA3 program package, Methods Mol. Biol. 132, 185–219.

Smith, T. F. and Waterman, M. S. (1981) Identification of common molecular subsequences, J. Mol. Biol. 47, 195–197.

Scoring Matrices

The Dayhoff (PAM) Scoring Matrices

Dayhoff, M. O., Barker, W. C., and Hunt, L. T. (1983) Establishing homologies in protein sequences, Methods Enzymol. 91, 534–545.

The BLOSUM Scoring Matrices

Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. USA 89, 10,915–10,919.

Henikoff, S. and Henikoff, J.G. (1991) Automated assembly of protein blocks for database searching, Nucleic Acids Res. 19, 6565–6572.

Fast Local Alignment Methods

Wootton, J. C. and Federhas, S. (1996) Analysis of compositionally biased regions in sequence databases, Methods Enzymol. 266, 554–571.

28 GCG Database Searching

David J. Heard

Introduction

The various programs available for searching GCG databases using text-based or sequence-based searches and the programs that allow the user create their own flat-file databases using the Genetics Computer Group (GCG) and the X-windows SeqLab interface will be described. When appropriate the equivalent command line options will be listed using the same format as given in the GCG program manual, for example —*WORDsize = 2*. It will also be assumed that the reader has access to GCG help documents either through the SeqLab or SeqWeb interfaces or the GCG Program Manuals that come with the software package.

Most users of GCG are well acquainted with at least one or two of the many programs available to search flat-file databases by keyword (Stringsearch, Lookup) or with query sequences (BLAST, FastA). But many people are confused as to the differences between programs and how to choose the best application for their search. For example BLAST seems to be the default program for most sequence-based searches because it is fast, but it may not always be the correct choice. We will review the pros and cons of each search program, how best to use them and how to interpret the output. A little known area of the GCG package will be described; i.e., the programs that allow scientists to create their own searchable databases using proprietary sequence data or extracts of public databases.

What Are Databases?

The definition of a database is simply a collection of records of similar content that can be easily accessed and managed. But there are two main types of databases, flat-file and relational. It is important to understand the difference between them and how this relates to GCG. Briefly, a flat-file is a file containing a number of records, or pieces of data that are not related to one another in a structured way. A typical GenBank record contains several pieces of information, such as the gene locus, accession number, organism name, the authors names, publications, features (i.e., the cds or coding sequence), and finally the DNA sequence itself. The GenBank flat-file database could theoretically be made up of a single large file containing several million sequence records (it is actually made of several large files containing different divisions of the

```
LOCUS       BH021105      191 bp    DNA             GSS       20-JUN-2001
DEFINITION  SP6 RPCI-1 55N16 RPCI-1 PAC library Homo sapiens genomic, DNA
            sequence.
ACCESSION   BH021105
VERSION     BH021105.1  GI:14495569
KEYWORDS    GSS.
SOURCE      human.
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 191)
  AUTHORS   Van Roy,N., Vandesompele,J., Berx,G., Staes,K., Van Gele,M., De
            Smet,E., De Paepe,A., Laureys,G., Versteeg,R., Van Roy,F. and
            Speleman,F.
  TITLE     Localisation of the 17q breakpoint of a constitutional 1;17
            translocation in a patient with neuroblastoma within a 50 kb
            segment located between the ACCN1 and TLK2 genes and near the
            distal breakpoints of two microdeletions in NF1 patients
  JOURNAL   Unpublished (2001)
COMMENT     Contact: Nadine Van Roy
            Dept Medical Genetics
            University Hospital Ghent
            De Pintelaan 185, 9000 Ghent, Belgium
            Tel: 3292405518
            Fax: 3292404970
            Email: nadine.vanroy@rug.ac.be
            Plate: 55  row: N  column: 16
            Seq primer: SP6
            Class: PAC end
            High quality sequence stop: 190.
FEATURES             Location/Qualifiers
     source          1..191
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
                     /map="17"
                     /clone_lib="RPCI-1 PAC library"
BASE COUNT       47 a     33 c     62 g     43 t      6 others
ORIGIN
        1 ccctttgaan cccgttcngg cccngaatgt aggggtagag gagaggcagg ggacgctgct
       61 tgacctggag tgagggtcag tgagttttct aaatagctgt caataaattg tctgctgact
      121 gatatatcag ggtcaaggag aggacagcga ttctcgtacg aacggttacg attcgagaga
      181 ggngagagnt n
//
LOCUS       BH021106      359 bp    DNA             GSS       20-JUN-2001
DEFINITION  SP6 RPCI-1 266J21 RPCI-1 PAC library Homo sapiens genomic, DNA
            sequence.
ACCESSION   BH021106
VERSION     BH021106.1  GI:14495570
KEYWORDS    GSS.
SOURCE      human.
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 359)
  AUTHORS   Van Roy,N., Vandesompele,J., Berx,G., Staes,K., Van Gele,M., De
            Smet,E., De Paepe,A., Laureys,G., Versteeg,R., Van Roy,F. and
            Speleman,F.
  TITLE     Localisation of the 17q breakpoint of a constitutional 1;17
            translocation in a patient with neuroblastoma within a 50 kb
            segment located between the ACCN1 and TLK2 genes and near the
            distal breakpoints of two microdeletions in NF1 patients
  JOURNAL   Unpublished (2001)
COMMENT     Contact: Nadine Van Roy
            Dept Medical Genetics
            University Hospital Ghent
            De Pintelaan 185, 9000 Ghent, Belgium
            Tel: 3292405518
            Fax: 3292404970
            Email: nadine.vanroy@rug.ac.be
            Plate: 266  row: J  column: 21
```

Fig. 1. The GenBank flat file database format. Each sequence record begins with the LOCUS and ends with a double slash (//) on a separate line. There are many thousands of records in the GenBank flatfiles.

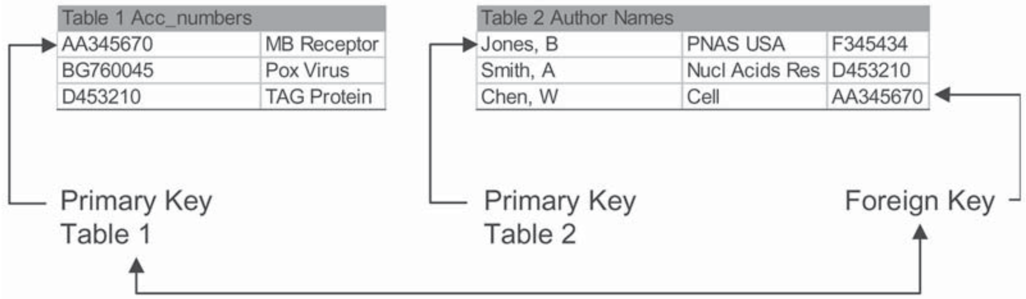


Table 1 Acc_numbers	
AA345670	MB Receptor
BG760045	Pox Virus
D453210	TAG Protein

Table 2 Author Names		
Jones, B	PNAS USA	F345434
Smith, A	Nucl Acids Res	D453210
Chen, W	Cell	AA345670

Fig. 2. An example of a simple relational database showing two tables and how they are related to one another. The foreign key in Table 2 of this figure relates this information to the information in Table 1 of this figure.

database). The individual records are separated from each other by a simple character (in GenBank a double slash, //, on a separate line), which indicates the end of one sequence record and the beginning of the next (*see* Fig. 1).

In contrast to flat-file databases, relational databases are made up of sets of formally described tables that are comprised of one or more columns containing different categories of data, i.e., species name or accession number, and rows of data for each unique record. The tables are related (hence the name relational) to one another by shared data fields called keys. For example in Fig. 2, two tables of a simple relational database are depicted. Table 1 of Fig. 2 contains accession numbers and sequence names and the key piece of information in this table is the accession number. Therefore this is called the *primary* key of that table. In Table 2 of Fig. 2, which contains author and journal names, the primary key is the author name. However, this table also contains a list of the accession numbers associated with each author. In Table 2 the accession number is called a *foreign* key and it is this information that *relates* the two tables.

The difference between the types of databases is obvious when you want to retrieve data from them. If one wanted to search the GenBank flat-file database for a specific gene one might have to search through every sequence record in the whole database, which could take days to finish. In contrast, the data in a relational database can be accessed and assembled in many different ways using a special programming language, the Structured Query Language (SQL). Using SQL queries one can limit the search to the tables containing the desired information, i.e., sequence names, ignoring all other information. However, it is important to note that while a relational database is more powerful and flexible than a flat-file database, it is much more complicated to construct, maintain, and query, and requires more computer resources. The standard release of the GCG program uses only flat-file databases. A special version of GCG was just released that can also interact with relational databases.

Searching By Keyword

Different pieces of data in a flat-file database can be extracted into separate files and individually searched. This process, called *database indexing*, makes flat-file database searching much faster and more efficient, providing many of the advantages of a relational database without the requirement of large amounts of system

Table 1
Relative Speed of the GCG Text-Based Search Tools Indicating Search Time and Results of a Search for the String: Estrogen Related Receptor[a]

Program	*Record searched*	*Time (hh:mm:ss)*	*Result of search*
Stringsearch	Definitions	00:00:27	Negative
Stringsearch	Full annotations	00:27:07	Negative
Lookup	Full annotations	00:00:06	Positive

[a] Performance on other machines will vary.

resources or knowledge of SQL. GCG contains a number of programs that perform database indexing and these are used to create the files searched by the GCG text-based search tools.

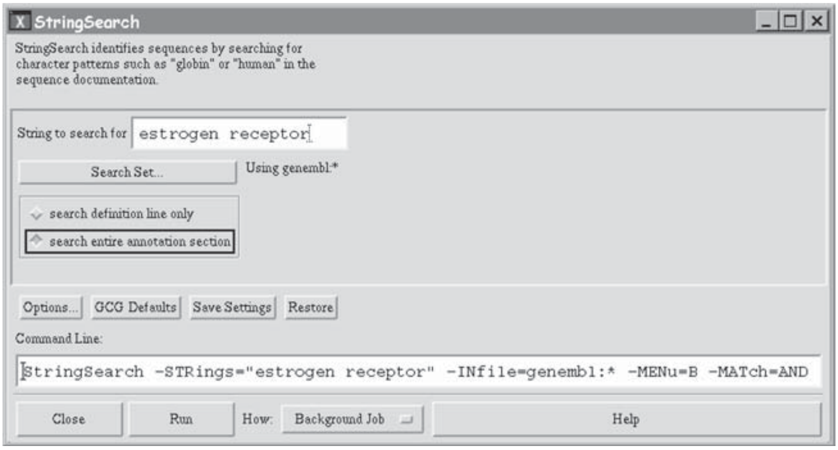
Stringsearch

To work with the GCG package, the GenBank flat-file database must be reformatted using one of the GCG database reformatting utilities (GenBanktoGCG) resulting in the sequence and the annotation information being split into different files called genbank.seq (sequence) and genbank.ref (annotation). Another program, Seqcat, indexes the database by extracting the locus name, accession number, definition line, date of submission and sequence length from genbank.ref into a genbank.seqcat file.

The GCG program Stringsearch performs simple text-based searches on the *dbname*.ref and *dbname*.seqcat files using queries (strings) typed by the user. In the SeqLab Stringsearch window you have the option of choosing to search the definition line contained in the *dbname*.seqcat file (—MENu=A) or the entire sequence annotation section in *dbname*.ref (—MENu=B). While searching the definition line alone is fast, only a limited amount of information is available. In contrast the entire sequence annotation may contain much more information but search times are many times longer (*see* Table 1).

The default setting in the program is to match **ALL** the words in the search string using the Boolean operator **AND** (—MATch=AND). In the example in Fig. 3, only records containing the words estrogen AND receptor would be returned. By selecting the **ANY** option (—MATch=OR) all records containing either estrogen **OR** receptor will be identified. Searches can be written in specific ways to make them more or less stringent. Wildcards (*), which allows one to search all spellings of a word, can make searches less stringent. For example, **hum*** would find entries with the words **hum**an or **hum**ans but may also find those containing the word **hum**mingbird. Quotation marks can be used to limit searches to specific word orders. A query for "*estrogen receptor*" would only find entries containing the words in the same order as typed and containing a space between them but would not find for example *estrogen-receptor* or *receptor estrogen*.

The output of the Stringsearch program is a GCG list file. This file can be imported into the SeqLab main or editor window enabling the user to access each sequence in the list through this interface. Alternatively, a list file can be used as input to other files including Stringsearch, BLAST, and FastA on the command line.



StringSearch identifies sequences by searching for character patterns such as "globin" or "human" in the sequence documentation.

String to search for: estrogen receptor

Using genembl:*

○ search definition line only
● search entire annotation section

Command Line:

```
StringSearch -STRings="estrogen receptor" -INfile=genembl:* -MENu=B -MATch=AND
```

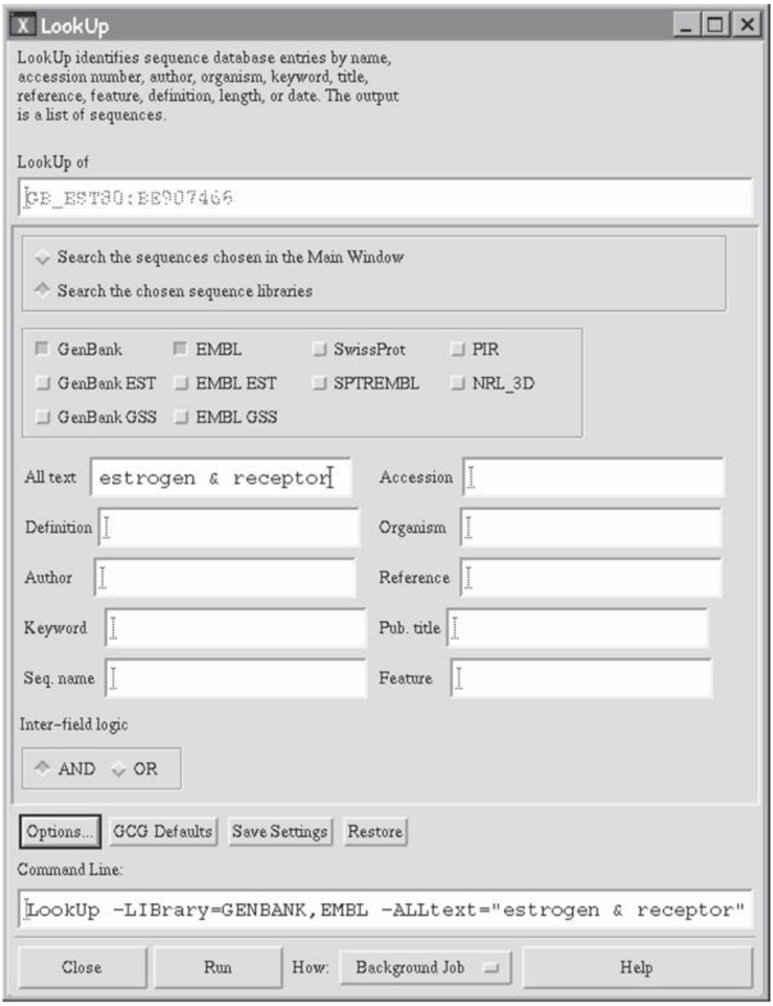
How: Background Job

Fig. 3. The SeqLab Stringsearch window. Queries are typed or pasted into the *String to search for* window, (e.g., estrogen receptor). The database or list of sequences to search is chosen by clicking **Search Set...**. One can search only the definition or the entire sequence annotation. As in all SeqLab program windows, the various command line options can be chosen by clicking on the **Options** button. The **Help** button gives access to the complete program description and access to all other GCG help pages. The actual command as it would appear on the command line is indicated at the bottom of the window.

Lookup

The GCG package contains a suite of indexing programs, the Sequence Retrieval System (SRS) programs written for GCG by Etzold and Argos (1993) that extract and build the SRS indices used by the *Lookup* program. The different pieces of information extracted by SRS include: sequence name, accession number, definition, organism, keywords, author names, publication title, reference, features, sequence length, and date of entry into the database. Each individual piece of information is placed into a separate file so that each can be individually searched. Note that the complete SRS program in its present form (version 6) is an extremely powerful database search tool that is freely available on the internet (*see* Website: http://srs.ebi.ac.uk/). In order for *Lookup* to access a database it must have a format that is recognized by the SRS indexing software otherwise it will not be useful. (This is described in detail in the GCG System Support Manual).

Lookup is a much more powerful and faster text-based search program than Stringsearch (*see* Fig. 4). To gain full advantage of Lookup, it should be accessed from the SeqLab interface, as it is a little clumsy from the command line. The main advantage of Lookup compared to Stringsearch is that one can write database queries using boolean operators AND (**&**), OR (**|**), and BUT-NOT (**!**) in the same query allowing the creation of very complex queries. An example of a complex query might be: **(mrna | cdna | (complete & cds)) ! (promoter | *pseudogene* | *intron***). This is interpreted as "find entries containing the words mRNA or cDNA or complete cds but exclude those containing promoter or pseudogene in any spelling (including retropseudogene, pseudogenes or retro-pseudogenes) or intron in any spelling" (alternatively in bold or italics). Frequently typed queries can be saved in text files and copied and pasted into the appropriate fields on the interface. Another important aspect of



LookUp identifies sequence database entries by name, accession number, author, organism, keyword, title, reference, feature, definition, length, or date. The output is a list of sequences.

LookUp of: GB_EST30:BE907466

○ Search the sequences chosen in the Main Window
● Search the chosen sequence libraries

☒ GenBank ☒ EMBL □ SwissProt □ PIR
□ GenBank EST □ EMBL EST □ SPTREMBL □ NRL_3D
□ GenBank GSS □ EMBL GSS

All text: estrogen & receptor
Accession:
Definition:
Organism:
Author:
Reference:
Keyword:
Pub. title:
Seq. name:
Feature:

Inter-field logic: ● AND ○ OR

Command Line:

```
LookUp -LIBrary=GENBANK,EMBL -ALLtext="estrogen & receptor"
```

How: Background Job

Fig. 4. The SeqLab Lookup interface allows you to click on one or all the databases that are formatted by the SRS tools. One can also limit the search to a group of sequences chosen in the main window (for example a List file of BLAST results). Text searches can be limited to certain parts of the annotation, (e.g., Author name) or to all the text in the annotation (All text). The Inter-field logic (AND/OR) buttons near the bottom of the window indicate the relationship between the different fields (e.g., Accession number: AF00* AND Organism: Homo sapiens).

Lookup is the ability to limit the search to specific parts of the sequence annotation, including organism, authors, and features. However, not all databases contain the same types of annotation. On occasion this can cause problems when searching different databases. One can routinely parse the GenBank database updates for specific subsets of genes that are of specific interest using Lookup with saved complex queries. For example, all the mammalian cDNAs that represent the complete coding sequence of a gene can be separated from all the other sequences using this tool. Alternatively one can parse the database for all sequences from a particular species (e.g., rabbit EST sequences) in one step.

The output from a Lookup query is a GCG list file, which can be imported into the main window or used in other GCG programs. This feature permits list refinement on a previous Lookup result, rather than searching the entire database again. For example, in a first Lookup query you asked for *all human EST sequences*. By importing the output of this search into the main window you can use Lookup again to find only those ESTs that are associated with prostate cancer. This combination of complex queries and list refinement make Lookup an extremely powerful database search and retrieval tool.

Lookup has a few disadvantages. One of the main drawbacks is that the files containing the indices require a substantial amount of disk space, for the full GCG database release of June 2001 the SRS indices require nearly 8 gigabytes of disk space. Also the indices are not as straightforward to create as those for Stringsearch and some databases will be incompatible with the SRS indexing software. However, as the example below will indicate, the performance of Lookup is superior compared to Stringsearch.

The example search summarized in Table 1 should illustrate the difference between Stringsearch and Lookup. A search for the string estrogen-related receptor was performed with both Stringsearch and Lookup using the Unix server with the aim of finding the accession number of the estrogen related receptor gamma (ERRgamma) gene. The Stringsearch of *definitions* in the GenEMBL database returned a result in 27 s but only found two entries for estrogen related receptor *alpha*, therefore neither was correct. A Stringsearch of the **entire** GenEMBL annotations with the same query took 27 min (or 60 times longer) and returned the same two genes. In contrast, using Lookup searched the **entire** GenEMBL annotations and returned a result in 6 s (270 times faster than Stringsearch) yielding 20 matches, amongst which was the gene of interest.

Searching with Query Sequences

GCG version 10.2 SeqLab contains the following sequence search programs: BLAST, NetBLAST, FastA, Ssearch, TFastA, TFastX, FastX, FrameSearch, HMMerSearch, MotifSearch, ProfileSearch, FindPatterns, Motifs, and WordSearch (*see* Fig. 5). The GCG package can be described as a toolbox, packed with tools of various shapes and sizes. The trick is to know which tool you need for the job. As shown in Table 2, the search tools can be divided into 3 basic types based on the type of search. In general BLAST, NetBLAST, (T)FastA/X, WordSearch, Ssearch, and FrameSearch are used to search databases with a single query sequence in order to find entries with similarity to the input sequence.

The programs HMMerSearch, MotifSearch, and ProfileSearch are similar in that groups of related sequences are first aligned to identify conserved motifs and the alignments are used to search the database for related sequences. Finally, Findpatterns and Motifs use simple word matching algorithms to find matches in short sequence patterns rather than searching full sequence databases. These will not be discussed further in this chapter.

It is important to understand how the search programs find and evaluate hits. The concept of scoring matrices is an important part of this task. A scoring matrix is a table that attempts to put a value on identities and substitutions in an alignment. For nucleotide-nucleotide comparisons the scoring matrices are rather simple. In BLAST any match is given a score of +1 and a mismatch of –3 by default but these numbers can be changed at the command line (—MATCH=2 and —MISmatch=5) or in the BLAST

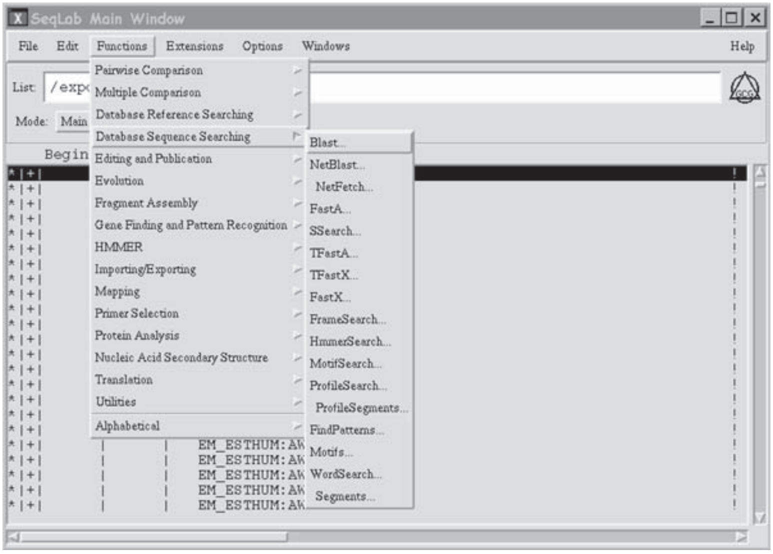


Fig. 5. The SeqLab main window showing all the Database Sequence Searching programs.

options window when using the SeqLab interface. In protein-protein comparisons scoring matrices are more complicated. When making an alignment the match or mismatch value of each alignment pair is obtained from a probabilistic model that considers the evolutionary conservation of amino acids at each position in related proteins. The original scoring matrices were determined by globally aligning highly conserved (85% identical) protein sequences from different organisms and determining the probability and type of amino acid substitutions in each position resulting in the percentage acceptable mutations (PAM) matrices described by Dayhoff et al. (1978). Matrices for more distantly related protein sequences were then extrapolated from these original scores. There are several different PAM matrices including PAM40, PAM70, and PAM250. Each refers to the evolutionary distance expressed as the percentage of positions that would be expected to mutate over a period of 10^8 y. The greater the value is, the greater the evolutionary distance. Therefore, PAM250 indicates a matrix for aligning proteins where 250% of all positions (or 2.5 substitutions at each position) would have changed over 100 million years.

However, as the PAM matrices were derived using global alignments on closely related sequences they are probably not the best choice to identify highly conserved protein domains in otherwise weakly related proteins. This problem was addressed by Henikoff and Henikoff (1992), who developed a new series of matrices called BLOCKS substitution matrices (BLOSUM, pronounced blossom) by comparing the ungapped, highly conserved protein sequences in the BLOCKS database. All BLOSUM matrices from BLOSUM 30 to BLOSUM 100 were derived from actual protein-protein alignments rather than being extrapolated mathematically. BLOSUM matrices have been found to be more reliable than PAM for aligning members of protein families. The numbers of the different BLOSUM matrices indicate the percent identity of the protein blocks used to generate the matrices. For example, BLAST, Ssearch, and FrameSearch all use the BLOSUM62 matrix by default, which was gen-

Table 2
The Characteristics of the Various GCG Search Tools and Their Major Advantages and Disadvantages

Program	*Nuc-Nuc*	*Nuc-Pro*	*Pro-Pro*	*Pro-Nuc*	*Gapped alignments?*	*Major advantages*	*Major disadvantages*
BLAST	Yes	Yes	Yes	Yes	Yes	Speed, gapped alignments, automatic detection of type of search.	Requires large local databases in a specific format. SeqLab can only use one database at a time.
NetBLAST	Yes	Yes	Yes	Yes	Yes	Searches NCBI Database over the internet.	Sequences submitted over internet therefore insecure.
FastA	Yes	No	Yes	No	No	Possibly more sensitive than BLAST in nuc-nuc searches. Flexible as you can search small groups of sequences rather than whole databases.	Slow and poor at aligning sequences with small gaps let alone cDNA vs Genomic sequence.
TFastA	No	No	No	Yes	No	See TFastX.	See TFastX.
TFastX	No	No	No	Yes	Frame-shifts only	Allows frameshifts.	Extremely slow. Poor at aligning sequences with gaps.
FastX	No	Yes	No	No	No	Allows frameshifts.	Slow and poor at aligning sequences with small gaps. Insensitive compared to BLAST.
WordSearch	Yes	No	Yes	No	No	Rapid.	Very insensitive compared to BLAST.
Ssearch	Yes	No	Yes	No	Small local gaps	Greater sensitivity.	Very slow.
FrameSearch	No	Yes	No	Yes	Frame-shifts only	More sensitive than BLAST or FastX.	Slower than BLAST and FastX.
HMMerSearch	Yes	No	Yes	No	No	Sensitive for finding weakly related sequences.	Slow, requires a number of related genes to build a profile.
MotifSearch	Yes	No	Yes	No	No	Useful for finding novel motifs in protein families.	Requires a number of related genes to build a profile.
ProfileSearch	No	No	Yes	No	No	Can find sequences with weak homology to the profile.	Slow, requires a number of related genes to build a profile. Not as sensitive as HMMerSearch.
FindPatterns	Yes	No	Yes	No	No	Finds short sequence patterns like restriction enzyme or proteolytic cleavage sites.	Restricted to short sequence patterns no database searching.
Motifs	No	No	Yes	No	No	Searches PROSITE database of protein sites and patterns.	Only finds motifs present in PROSITE cannot find novel motifs.

erated with BLOCKS with up to 62% identity. In principle lower numbered, e.g., BLOSUM40 matrices should be used with weakly conserved proteins and higher numbered, e.g., BLOSUM80 matrices with more conserved proteins.

An amino acid identity between query and subject gives a relatively large positive score: e.g., +4 (in BLOSUM62; *see* Fig. 6). A conservative amino acid substitution, such as leucine to isoleucine, results in a smaller positive score for that position (+2), whereas a nonconservative change, for example leucine to aspartate, results in a large negative score (–4). Changes that result in substitutions of hydrophobic for hydrophilic amino acids are penalized more than changes of amino acid polarity. The insertion of a gap has a large negative score (–8) by default in BLOSUM62 whereas the extension of a gap is penalized to a lower degree (–2 for each amino acid skipped). These numbers are then used to calculate the best alignment of the query sequence to that found in the database and to generate the sequence score. A description of all the scoring matrices available in GCG and how to manipulate them is found in the appendix VII of the GCG program manual.

BLAST/NetBLAST

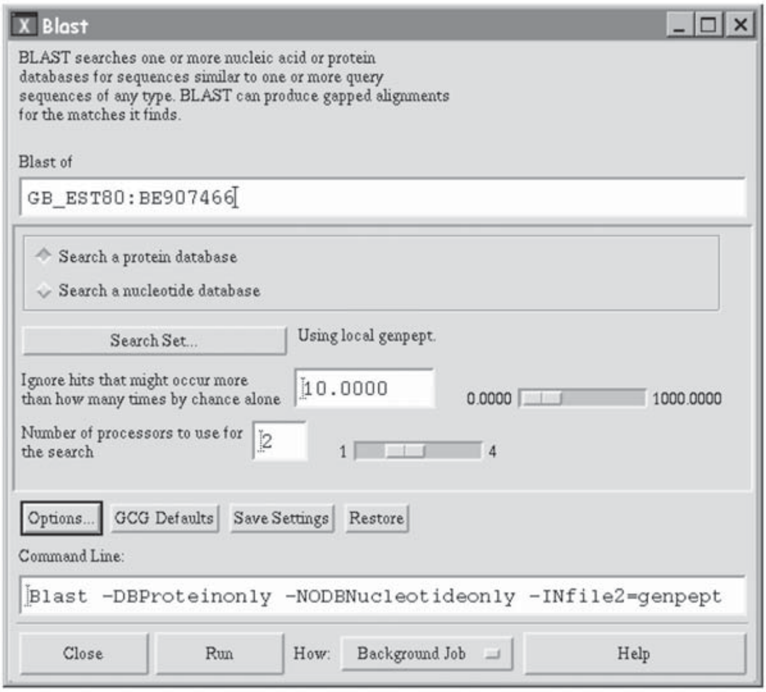
The Basic Local Alignment Search Tool (BLAST) algorithm was described by Altschul, et al. (1990) and quickly became popular primarily because of its speed in comparison to other search algorithms. The current implementation of BLAST, called gapped BLAST or BLAST2 (Altschul, et al., 1990), was included in GCG beginning in version 10.0 (*see* Fig. 7A). This implementation allows gapped sequence alignments, meaning that more than one alignment is possible for each entry in the database. BLAST2 is approx three-fold faster than the original BLAST program and is faster than all other database searching tools (*see* Table 3). In our toolbox analogy BLAST2 is the handy adjustable wrench in your toolbox; if you could only have one wrench in the toolbox this would be the one to have. BLAST works well with several different types of databases from DNA to protein and EST to Genomic DNA. It rapidly answers the question: Which sequences in the database are related to my sequence, regardless of length, presence of large insertions or deletions, and sequence type. However, for serious protein function prediction BLAST is probably not the tool of choice.

NetBLAST is essentially identical to BLAST except NetBLAST uses the internet to query the databases maintained at the National Center for Biotechnology Information (NCBI). It is not recommended that you use NetBLAST with query sequences that you want to keep confidential. Be aware that sending a sequence on an insecure line may be considered public disclosure. If this is not a concern then there are two main advantages of using NetBLAST. The first is that the databases at NCBI are updated on a daily basis so you can query the most up-to-date sequence data. This is important as the public sequence databases are growing very rapidly. For example, the human EST database grew by over 14% in the 2 mo period between GenBANK release 121 and 122. This translates to the depositioned 391,652 human ESTs, an average of 6,528 sequences/day. The second advantage to NetBLAST is that you will not have to invest in disk space to have the public sequence data in-house.

BLAST is actually 5 different programs: BLASTN for nucleic acid queries of nucleic acid databases; BLASTP for protein queries of protein databases; BLASTX for translated nucleic acid queries of protein databases; TBLASTN for protein queries of translated nucleic acid databases, and finally TBLASTX for translated nucleic acid queries of translated nucleic acid databases. The decision as to which BLAST

```
!!AA_SCORING_MATRIX_RECT 1.0

BLOSUM62 amino acid substitution matrix.

Reference: Henikoff, S. and Henikoff, J. G. (1992). Amino acid
           substitution matrices from protein blocks.  Proc. Natl. Acad.
           Sci. USA 89: 10915-10919.

                    February 20, 1996 14:33   ..

{
GAP_CREATE 8
GAP_EXTEND 2
}

     A   B   C   D   E    F   G   H   I   K    L   M   N   P   Q    R   S   T   V   W    X   Y   Z   *
A    4  -2   0  -2  -1   -2   0  -2  -1  -1   -1  -1  -2  -1  -1   -1   1   0   0  -3    0  -2  -1  -4
B   -2   4  -3   4   1   -3  -1   0  -3   0   -4  -3   3  -2   0   -1   0  -1  -3  -4   -1  -3   1  -4
C    0  -3   9  -3  -4   -2  -3  -3  -1  -3   -1  -1  -3  -3  -3   -3  -1  -1  -1  -2   -2  -2  -3  -4
D   -2   4  -3   6   2   -3  -1  -1  -3  -1   -4  -3   1  -1   0   -2   0  -1  -3  -4   -1  -3   1  -4
E   -1   1  -4   2   5   -3  -2   0  -3   1   -3  -2   0  -1   2    0   0  -1  -2  -3   -1  -2   4  -4

F   -2  -3  -2  -3  -3    6  -3  -1   0  -3    0   0  -3  -4  -3   -3  -2  -2  -1   1   -1   3  -3  -4
G    0  -1  -3  -1  -2   -3   6  -2  -4  -2   -4  -3   0  -2  -2   -2   0  -2  -3  -2   -1  -3  -2  -4
H   -2   0  -3  -1   0   -1  -2   8  -3  -1   -3  -2   1  -2   0    0  -1  -2  -3  -2   -1   2   0  -4
I   -1  -3  -1  -3  -3    0  -4  -3   4  -3    2   1  -3  -3  -3   -3  -2  -1   3  -3   -1  -1  -3  -4
K   -1   0  -3  -1   1   -3  -2  -1  -3   5   -2  -1   0  -1   1    2   0  -1  -2  -3   -1  -2   1  -4

L   -1  -4  -1  -4  -3    0  -4  -3   2  -2    4   2  -3  -3  -2   -2  -2  -1   1  -2   -1  -1  -3  -4
M   -1  -3  -1  -3  -2    0  -3  -2   1  -1    2   5  -2  -2   0   -1  -1  -1   1  -1   -1  -1  -1  -4
N   -2   3  -3   1   0   -3   0   1  -3   0   -3  -2   6  -2   0    0   1   0  -3  -4   -1  -2   0  -4
P   -1  -2  -3  -1  -1   -4  -2  -2  -3  -1   -3  -2  -2   7  -1   -2  -1  -1  -2  -4   -2  -3  -1  -4
Q   -1   0  -3   0   2   -3  -2   0  -3   1   -2   0   0  -1   5    1   0  -1  -2  -2   -1  -1   3  -4

R   -1  -1  -3  -2   0   -3  -2   0  -3   2   -2  -1   0  -2   1    5  -1  -1  -3  -3   -1  -2   0  -4
S    1   0  -1   0   0   -2   0  -1  -2   0   -2  -1   1  -1   0   -1   4   1  -2  -3    0  -2   0  -4
T    0  -1  -1  -1  -1   -2  -2  -2  -1  -1   -1  -1   0  -1  -1   -1   1   5   0  -2    0  -2  -1  -4
V    0  -3  -1  -3  -2   -1  -3  -3   3  -2    1   1  -3  -2  -2   -3  -2   0   4  -3   -1  -1  -2  -4
W   -3  -4  -2  -4  -3    1  -2  -2  -3  -3   -2  -1  -4  -4  -2   -3  -3  -2  -3  11   -2   2  -3  -4

X    0  -1  -2  -1  -1   -1  -1  -1  -1  -1   -1  -1  -1  -2  -1   -1   0   0  -1  -2   -1  -1  -1  -4
Y   -2  -3  -2  -3  -2    3  -3   2  -1  -2   -1  -1  -2  -3  -1   -2  -2  -2  -1   2   -1   7  -2  -4
Z   -1   1  -3   1   4   -3  -2   0  -3   1   -3  -1   0  -1   3    0   0  -1  -2  -3   -1  -2   4  -4
*   -4  -4  -4  -4  -4   -4  -4  -4  -4  -4   -4  -4  -4  -4  -4   -4  -4  -4  -4  -4   -4  -4  -4   1
```

Fig. 6. The Blosum62 protein-protein scoring matrix indicating amino acid identity and substitution scores. Amino acids are indicated along the top and left side by their single letter codes.



BLAST searches one or more nucleic acid or protein databases for sequences similar to one or more query sequences of any type. BLAST can produce gapped alignments for the matches it finds.

Blast of

GB_EST80:BE907466

◆ Search a protein database

◇ Search a nucleotide database

Search Set... Using local genpept.

Ignore hits that might occur more than how many times by chance alone 10.0000

Number of processors to use for the search 2

Command Line:

```
Blast -DBProteinonly -NODBNucleotideonly -INfile2=genpept
```

Fig. 7. **(A)** The BLAST window. In this example a DNA sequence is being used to search a protein database (genpept). The Ignore Hits slider changes the —EXPect variable, which determines the cutoff to show hits that would have occurred by chance alone (a higher number means more likely to occur by chance, or less significance). The number of processors slider tells the program how many processors that you are allowed to use for your search. Using more speeds up the program but slows down other applications running on the same computer. **(B)** *(opposite page)* The BLAST output in text format. Notice the line at the top of the output: *!!Sequence_List 1.0* —this tells GCG that this output is a valid list file that can be used to import the sequences into the SeqLab main window (note that this line appears in the output from all the GCG search programs described in this chapter). The score in bits and E-value scores appear on the right side at the top of the sequence list. An E-value of 0.0 means it is infinitely unlikely that this match occurred by chance alone. In the alignment portion the definition line of the database sequence is given followed by the scores and the orientation of the query and subject (database hit) strands. Plus equals forward direction, as the query sequence was entered into BLAST or the database sequence submitted to the database whereas "Minus" would indicate the reverse complement.

program to use is made automatically according to the type of input sequence (protein or nucleic acid) being used to search the type of database chosen. When using nucleotide sequences to query a protein database (BLASTX), the query sequence is translated in all six frames (3 forward and 3 reverse) and each resulting protein sequence is used to search the database. When using a protein sequence to query a DNA database (TBLASTN), the entire database is translated in all 6 frames and compared to the query peptide. The only BLAST program that has to be selected specifically by the user is TBLASTX. This program translates both the nucleic acid query sequence and the nucleic acid database in all 6 frames meaning that for each query sequence 36 different BLASTP searches must be performed. Note that TBLASTX is not capable of performing gapped alignments.

B

```
!!SEQUENCE_LIST 1.0
BLASTN 2.1.3 [Apr-1-2001]


Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schaffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Query= GB_PR:AB001636
         (3028 letters)

Database: genembl
          1,346,493 sequences; 3,479,015,987 total letters

Searching. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . .done

                                                        Score     E
Sequences producing significant alignments:             (bits)  Value ..

GB_PR:AB001636  Begin: 342 End: 2999
!AB001636 Homo sapiens mRNA for ATP-dependent RNA ...      5269  0.0
GB_PR:AB001636  Begin: 1 End: 296
!AB001636 Homo sapiens mRNA for ATP-dependent RNA ...
GB_PR:AF279891  Begin: 181 End: 2441
!AF279891 Homo sapiens dead box protein 15 mRNA, c...      4365  0.0
GB_PR:AF279891  Begin: 1 End: 135
!AF279891 Homo sapiens dead box protein 15 mRNA, c...
GB_RO:BC003745  Begin: 323 End: 2971
!BC003745 Mus musculus, Similar to DEAD/H (Asp-Glu...      3897  0.0
GB_RO:BC003745  Begin: 129 End: 273
!BC003745 Mus musculus, Similar to DEAD/H (Asp-Glu...
GB_RO:BC003745  Begin: 13 End: 42
!BC003745 Mus musculus, Similar to DEAD/H (Asp-Glu...
...
GB_PR:AB001601  Begin: 1482 End: 1524
!AB001601 Homo sapiens DBP2 mRNA for ATP-dependent...        62  2e-06
GB_PAT:AX013689  Begin: 157 End: 199
!AX013689 Sequence 9 from Patent WO9954460. 9/2000           62  2e-06
GB_PL:NCB2O8  Begin: 43364 End: 43436
!AL355930 Neurospora crassa DNA linkage group II BAC...      50  0.009
GB_IN:CET21B6  Begin: 20347 End: 20369
!Z68011 Caenorhabditis elegans cosmid T21B6, comple...       46  0.14
GB_IN:CELC02D5  Begin: 9310 End: 9332 Strand:-
!L16622 C. elegans cosmid C02D5. 10/1993                     46  0.14

\\End of List

>GB_PR:AB001636 AB001636 Homo sapiens mRNA for ATP-dependent RNA
            helicase #46, complete cds. 12/1997
          Length = 3028

 Score = 5269 bits (2658), Expect = 0.0
 Identities = 2658/2658 (100%)
 Strand = Plus / Plus


Query: 342   ttgcgagcttcaacaaatgctatgcttatcagtgctggattaccacccctgaaagcttcc 401
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 342   ttgcgagcttcaacaaatgctatgcttatcagtgctggattaccacccctgaaagcttcc 401


Query: 402   cattcagctcactcaacccactcagcacattcaacgcattctacacattctgctcattca 461
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct: 402   cattcagctcactcaacccactcagcacattcaacgcattctacacattctgctcattca 461
```

Table 3
Relative Speed of the GCG Query Sequence Search Tools

Search Type	*Program Name*	*Search Time (hh:mm:ss)*
Nuc-NucDB	BLASTN	00:00:31
Nuc-NucDB	BLASTN (wordsize=7)	00:12:38
Nuc-NucDB	FastA	00:29:32
Nuc-NucDB	SSearch	36:00:00[a]
NucX-TNucDB	TBLASTX	00:09:51
Nuc-ProDB	BLASTX	00:00:56
Nuc-ProDB	FastX	00:02:52
Nuc-ProDB	FrameSearch	00:41:17
Pro-ProDB	BLASTP	00:00:22
Pro-ProDB	FastA	00:02:29
Pro-ProDB	SSearch	00:08:12
Pro-NucDB	TBLASTN	00:04:58
Pro-NucDB	TFastX	01:39:03
Pro-NucDB	FrameSearch	36:00:00[a]

[a] Searched stopped due to drain on system resources. Performance on other machines will vary.

Nuc, GenBank Accession AB001636. NucDB, GenBank (release 123). NucX, AB001636 translated. TNucDB, translated GenBank (release 123). Pro, Swissprot Accession DD15_Human. ProDB, Swissprot (release 39).

The BLAST algorithm is complex (*see* the GCG manuals, Altschul, et al. (1997), and Chapter 27 for a detailed description). BLAST identifies words in the query sequence and searches for sequences with similar words in the database. The word size is the smallest region of 100% identity between a query sequence and the sequences in a database for a sequence to be identified as a potential hit. The default word size is 11 for nucleotide-nucleotide and 3 for protein-protein comparisons. This can be changed in the options menu or at the command line **(—WORDsize=n)**. When the program finds a similar word in a sequence in the database it tries to extend the alignment outwards in both directions creating a high scoring segment pair (HSP). Once an HSP is identified, the algorithm searches in both directions for other HSPs in the same sequence to generate the gapped alignments. It is often said that FastA is more sensitive in nucleotide-nucleotide searches than BLASTN. While this may be true using the default BLASTN settings, decreasing the wordsize to 7 results in BLAST being nearly as sensitive as FastA, but faster (*see* Table 3).

Unlike other GCG search algorithms, by default, BLAST filters or masks regions of low sequence complexity in the query sequence. Filtering removes a substantial amount of unwanted noise from the search output. A database search with a cDNA sequence containing a long poly-A stretch is a good example of the usefulness of a filter. Without filtering, or physically deleting the poly-A sequence, such a search would return thousands of statistically significant but uninteresting hits to the polyA region that could mask potentially interesting alignments. The filtering program for nucleic acid queries of nucleic acid databases (BLASTN) is called DUST and was written by Tatusov and Lipman (unpublished). DUST replaces low complexity nucleotide sequences with N's that are ignored by BLAST. All other BLAST programs use the SEG algorithm (Wootton and Federhen, 1996) to filter proteins for low complexity protein regions (for example stretches of glutamines or prolines) replacing those residues with an X (*see* the example in Fig. 8A). Note that filtering is only applied to the query sequence, not to database sequences. Filtering can be turned off when desired in both BLAST and NetBLAST. (For further discussion on the effects of filtering tools on database searching, *see* "Suggested Readings," Shpaer, et al.)

The BLAST output in GCG is essentially comprised of a text file containing a list of sequences and a number of alignments (*see* Fig. 7B). The length of the list and the number of alignments in the output can be controlled in the options menu **SeqLab** or at the command line **(—LIStsize=1000 —ALIgnments=500)**. The output also contains two different score values, which can aid evaluating aligned sequences. The first is called the bit score, which is presented as a whole number and represents the probability of finding the same alignment by chance. The bit score is the n in the formula 1/ 2^n. If the bit score is 30 then the chance of finding the same sequence randomly would be 1 in 2^{30} or 1 in a billion. A higher bit score is better than a lower one. Importantly, the bit score is independent of the size of the database searched. This can be important when comparing hits in a small database such as your personal sequence database and a large database like the GenEMBL ESTs. The other score given is the expected (E) value, which is also an expression of the probability of finding the same match by chance and is expressed as a number such as 5e-48 (5×10^{-48}). The E value is however dependent on the size of the database being searched. A lower E value score indicates a better match, with 0 being a perfect match.

FastA Suite/SSearch/FrameSearch

In the toolbox analogy the FastA/SSearch/Framesearch programs would be like a set of crescent wrenches; they come in all different sizes, both metric and imperial, and one has to find the right wrench for the job at hand. These tools are best for comparing protein sequences (or translated nucleotides) with protein databases (or translated nucleotide databases). They are better than BLAST at answering the question: What sequences in the database are distantly related to my input sequence?

The reason FastA and SSearch/Framesearch are grouped together is that they are slower, yielding more sensitive searches of the databases, yet similar output. However, these programs are different in terms of the algorithms used to find sequence matches. The FastA programs use the algorithm of Pearson and Lipman (1988), whereas SSearch and FrameSearch use the Smith and Waterman (SW) algorithm. The Pearson-Lipman and SW algorithms have been extensively compared using different scoring matrices and in most instances the SW algorithm appears to be more sensitive than Pearson and Lipman for protein-protein searches. However, FastA programs are much faster than SW (*see* Table 3). It is possible to search subsets of sequences rather than entire databases. This can dramatically increase performance if one is interested in a particular protein family.

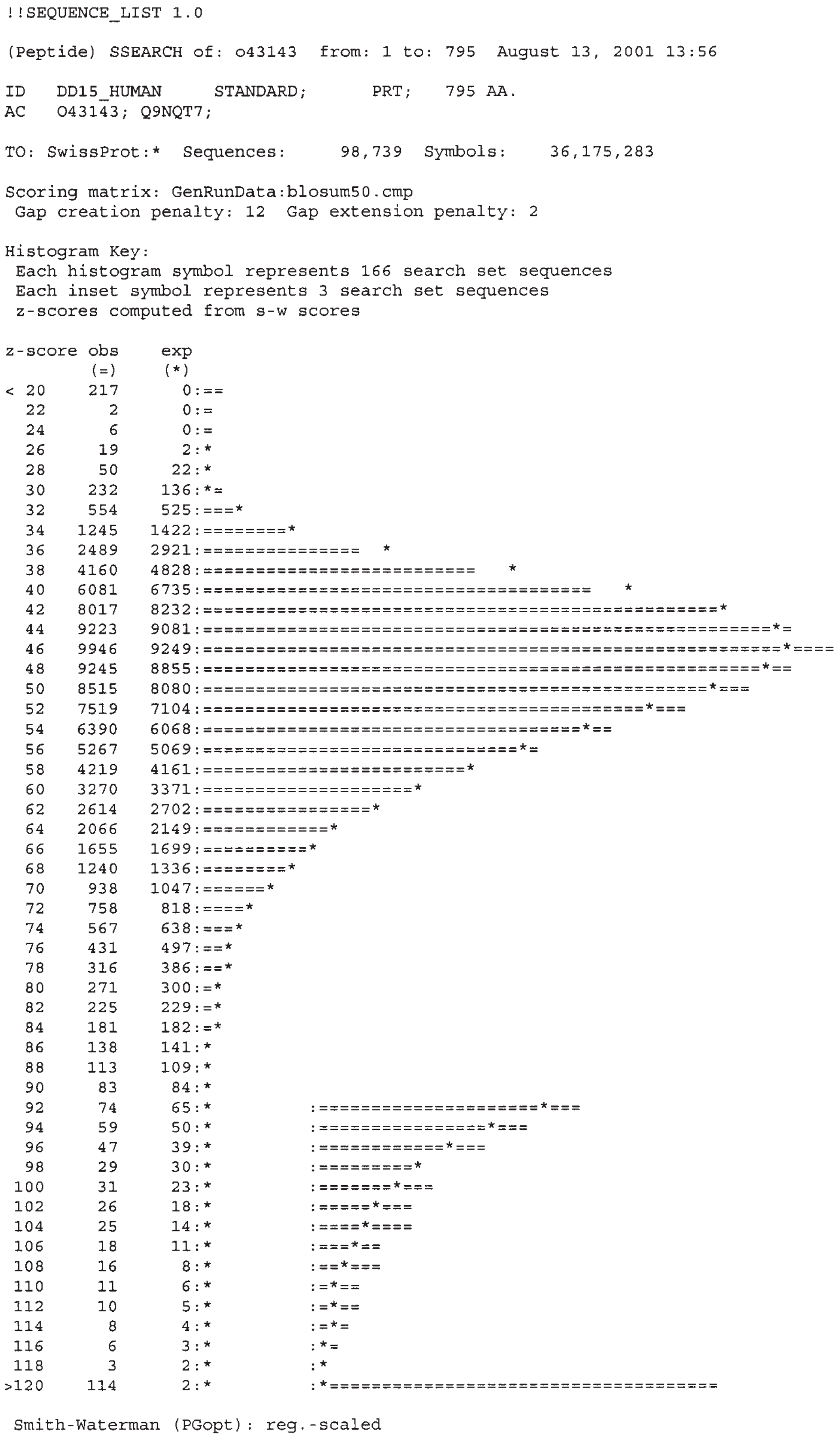
Like BLAST, FastA and Smith-Waterman (SW) searches come in several different flavors (*see* Table 3), specific to the type of search you want to execute. FastA and SSearch compare nucleotide queries against nucleotide databases and protein queries against protein databases. Both SSearch and FrameSearch are *extremely* slow when searching large databases for nucleotide-nucleotide comparisons (*see* Table 3). TFastA and TFastX compare protein queries against nucleic acid databases by translating the database sequences into all 6 frames (3 forward and 3 reverse). TFastX is probably more useful than TFastA in that it allows for frame-shifts in the alignment. Finally, FastX compares a nucleic acid query, translated in all 6 frames, against a protein database. FrameSearch is similar to a combined FastX and TfastX search in that a translated nucleotide query can be compared to a protein database or a protein query can be compared to a translated nucleotide database.

The output of the FastA programs and SSearch are quite similar. At the top of the output is a histogram showing the distribution of z-scores between the query sequence and the database sequences (*see* Fig. 8A) followed by the sequence list and alignments in the same file (*see* Fig. 8B). The histogram illustrates the number of z-scores expected (indicated by an asterisk) and the actual scores observed (bars of equal signs) for each value. This may be useful in some cases to determine whether the observed similarity is statistically significant. The histogram search results usually show a normal distribution, with a small number of very high scores at the bottom. The majority of alignments to database entries will be in the random noise at lower z-scores (the large peak). A second peak, inset at the bottom right, indicates the same region as it overlaps on the main histogram but at a lower scale (*see* the histogram key to see how many sequences each equals sign represents). It is the inset peak that one should examine for the presence of sequence entries with significant z-scores (i.e., the asterisk appears to the left of the equals sign). In summary, the histogram is a visual summation of whether the alignments that appear below in the output may be significant. In the example used the histogram indicates a slightly higher than expected number of hits in the z-score range from 44–56 and 92–116 (*see* Fig. 8A). This reflects the low complexity regions of the query which identified more sequences in the database than would be expected statistically. However, by manually filtering the complexity regions using the GCG programs Xnu or Seg prior to searching one could ensure that these low complexity regions do not mask the truly interesting relationships (*see* Fig. 8C).

The remaining portion of the output appears very much like a BLAST output with a list of sequences sorted by score followed by the sequence alignments (*see* Fig. 8B). There are 3 scores reported in the sequence list: 1) the s-w score (in SSearch) or opt score (in FastA) is the Smith-Waterman pairwise alignment score derived from the sum of the scoring matrix values at each position in the alignment. 2) The z-score, which is calculated by linear regression of the s-w score with the natural log (ln) of the length of the matching sequence from the database. 3) The E-value score for each sequence indicates the number of sequences that would have to be searched to give a z-score equal to the z-score obtained for that sequence match by random chance. It is similar to the E-value score described for BLAST. The output is sorted by the E-value in the sequence list and alignments. It is stated in the GCG program manual that for searches of a protein database of 10,000 sequences, an E-value of less than 0.01 indicates similarity to the query sequence and that E-values between 1 and 10 may also indicate relatedness (homotogy). The Smith-Waterman program does a better job of aligning two DNA sequences than FastA therefore it is suggested that the option **Use**

Fig. 8. The SSearch/FastA output. **(A)** *(opposite page)* The histogram showing the distribution of z-scores over all the data in the database. The equal-signs indicate the actual scores whereas the asterisks indicate the expected. Note that in this search the actual score is higher than the expected for most of the z-scores. This is because the sequence contains low complexity regions that were not screened out before the search (*see* **C**). In the inset histogram (bottom right) the higher z-score values are indicated at a lower scale than the main histogram (Main: 1equals-sign represents 166 database sequences. Inset 1 "=" represents 3 database sequences). *(Continued on pages 480 and 481)*

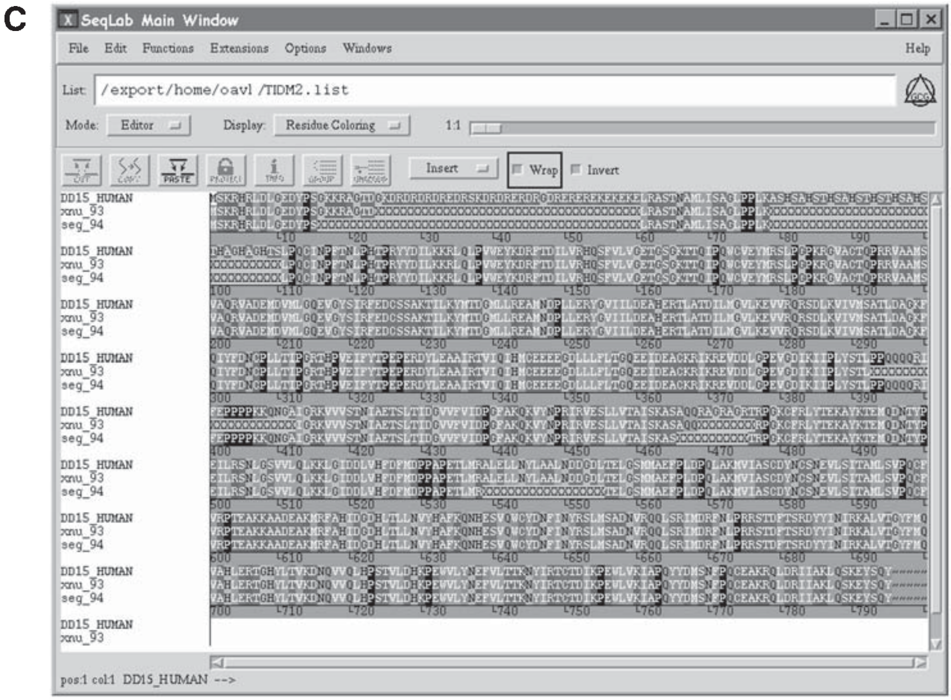
A



```
!!SEQUENCE_LIST 1.0

(Peptide) SSEARCH of: o43143  from: 1 to: 795  August 13, 2001 13:56

ID   DD15_HUMAN     STANDARD;      PRT;   795 AA.
AC   O43143; Q9NQT7;

TO: SwissProt:*  Sequences:     98,739  Symbols:    36,175,283

Scoring matrix: GenRunData:blosum50.cmp
 Gap creation penalty: 12  Gap extension penalty: 2

Histogram Key:
 Each histogram symbol represents 166 search set sequences
 Each inset symbol represents 3 search set sequences
 z-scores computed from s-w scores

z-score obs    exp
        (=)    (*)
< 20    217      0:==
  22      2      0:=
  24      6      0:=
  26     19      2:*
  28     50     22:*
  30    232    136:*=
  32    554    525:===*
  34   1245   1422:========*
  36   2489   2921:===============  *
  38   4160   4828:==========================   *
  40   6081   6735:=====================================   *
  42   8017   8232:=================================================*
  44   9223   9081:======================================================*=
  46   9946   9249:=======================================================*====
  48   9245   8855:=====================================================*==
  50   8515   8080:================================================*===
  52   7519   7104:==========================================*===
  54   6390   6068:====================================*==
  56   5267   5069:==============================*=
  58   4219   4161:=========================*
  60   3270   3371:====================*
  62   2614   2702:================*
  64   2066   2149:============*
  66   1655   1699:==========*
  68   1240   1336:========*
  70    938   1047:======*
  72    758    818:====*
  74    567    638:===*
  76    431    497:==*
  78    316    386:==*
  80    271    300:=*
  82    225    229:=*
  84    181    182:=*
  86    138    141:*
  88    113    109:*
  90     83     84:*
  92     74     65:*         :=====================*===
  94     59     50:*         :================*===
  96     47     39:*         :============*===
  98     29     30:*         :=========*
 100     31     23:*         :=======*===
 102     26     18:*         :=====*===
 104     25     14:*         :====*====
 106     18     11:*         :===*==
 108     16      8:*         :==*===
 110     11      6:*         :=*==
 112     10      5:*         :=*==
 114      8      4:*         :=*=
 116      6      3:*         :*=
 118      3      2:*         :*
>120    114      2:*         :*=====================================

 Smith-Waterman (PGopt): reg.-scaled
```

B

```
The best scores are:                                     s-w    z-sc  E(98464)..

SW:DD15_HUMAN     Begin: 1   End: 795
! O43143 homo sapiens (human). putati...                5316  5999.3       0
SW:DD15_MOUSE     Begin: 1   End: 758
! O35286 mus musculus (mouse). putati...                4992  5632.9       0
SW:DD15_CAEEL     Begin: 9   End: 732
! Q20875 caenorhabditis elegans. puta...                3548  3997.9  7e-216
SW:DD15_ARATH     Begin: 21   End: 717
! O22899 arabidopsis thaliana (mouse-...                3316  3735.3  2.9e-201
SW:PR43_YEAST     Begin: 71   End: 747
! P53131 saccharomyces cerevisiae (ba...                2914  3279.6  7.2e-176
SW:DD15_SCHPO     Begin: 54   End: 722
! O42945 schizosaccharomyces pombe (f...                2863  3222.2  1.1e-172
SW:DD15_STRPU     Begin: 1   End: 455
! O17438 strongylocentrotus purpuratu...                2666  3004.0  1.6e-160
SW:DDX8_HUMAN     Begin: 558   End: 1198
! Q14562 homo sapiens (human). atp-de...                2250  2522.9  1e-133
SW:DDX8_ARATH     Begin: 461   End: 1101
...
! P34594 caenorhabditis elegans. hypo...                 111   117.4     9.7
SW:POLG_TEV     Begin: 1241   End: 1529
! P04517 t genome polyprotein [contai...                 134   117.4     9.7
\\End of List


o43143
SW:DD15_HUMAN

ID   DD15_HUMAN     STANDARD;      PRT;   795 AA.
AC   O43143; Q9NQT7;
DT   15-DEC-1998 (Rel. 37, Created)
DT   15-DEC-1998 (Rel. 37, Last sequence update)
DT   20-AUG-2001 (Rel. 40, Last annotation update)
DE   PUTATIVE PRE-MRNA SPLICING FACTOR RNA HELICASE (DEAH BOX PROTEIN 15) . . .


SCORES   z-score: 5999.3 E(): 0
>>SW:DD15_HUMAN                                         (795 aa)
 Z-score: 5999.3 expect():    0
Smith-Waterman score: 5316;   100.0% identity in 795 aa overlap
 (1-795:1-795)

                     10        20        30        40        50        60
o43143       MSKRHRLDLGEDYPSGKKRAGTDGKDRDRDRDREDRSKDRDRERDRGDREREREKEKEKE
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
DD15_HUMAN   MSKRHRLDLGEDYPSGKKRAGTDGKDRDRDRDREDRSKDRDRERDRGDREREREKEKEKE
                     10        20        30        40        50        60

                     70        80        90       100       110       120
o43143       LRASTNAMLISAGLPPLKASHSAHSTHSAHSTHSTHSAHSTHAGHAGHTSLPQCINPFTN
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
DD15_HUMAN   LRASTNAMLISAGLPPLKASHSAHSTHSAHSTHSTHSAHSTHAGHAGHTSLPQCINPFTN
                     70        80        90       100       110       120

                    130       140       150       160       170       180
o43143       LPHTPRYYDILKKRLQLPVWEYKDRFTDILVRHQSFVLVGETGSGKTTQIPQWCVEYMRS
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
DD15_HUMAN   LPHTPRYYDILKKRLQLPVWEYKDRFTDILVRHQSFVLVGETGSGKTTQIPQWCVEYMRS
                    130       140       150       160       170       180

                    190       200       210       220       230       240
o43143       LPGPKRGVACTQPRRVAAMSVAQRVADEMDVMLGQEVGYSIRFEDCSSAKTILKYMTDGM
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
DD15_HUMAN   LPGPKRGVACTQPRRVAAMSVAQRVADEMDVMLGQEVGYSIRFEDCSSAKTILKYMTDGM
                    190       200       210       220       230       240

                    250       260       270       280       290       300
o43143       LLREAMNDPLLERYGVIILDEAHERTLATDILMGVLKEVVRQRSDLKVIVMSATLDAGKF
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
DD15_HUMAN   LLREAMNDPLLERYGVIILDEAHERTLATDILMGVLKEVVRQRSDLKVIVMSATLDAGKF
                    250       260       270       280       290       300
```



List: /export/home/oavl /TIDM2.list

Mode: Editor Display: Residue Coloring 1:1

```
DD15_HUMAN  MSKRHRLDLGEDYPSQKKRAGTDGKDRDRDRDREDRSKDRDRERDRGDREREREKEKEKELRASTNAMLISAGLPPLKASHSAHSTHSAHSTHSTHSAHS
xnu_93      MSKRHRLDLGEDYPSQKKRAGTDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXLRASTNAMLISAGLPPLKXXXXXXXXXXXXXXXXXXXXXX
seg_94      MSKRHRLDLGEDYPSXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXLRASTNAMLISAGLPPLKXXXXXXXXXXXXXXXXXXXXXX
                     └10       └20       └30       └40       └50       └60       └70       └80       └90       └
DD15_HUMAN  DHAGHAGHTSLPQCINPFTNLPHTPRYYDILKKRLQLPVWEYKDRFTDILVRHQSFVLVGETGSGKTTQIPQWCVEYMRSLPGPKRGVACTQPRRVAAMS
xnu_93      XXXXXXXXXXLPQCINPFTNLPHTPRYYDILKKRLQLPVWEYKDRFTDILVRHQSFVLVGETGSGKTTQIPQWCVEYMRSLPGPKRGVACTQPRRVAAMS
seg_94      XXXXXXXXXXLPQCINPFTNLPHTPRYYDILKKRLQLPVWEYKDRFTDILVRHQSFVLVGETGSGKTTQIPQWCVEYMRSLPGPKRGVACTQPRRVAAMS
            100      └110      └120      └130      └140      └150      └160      └170      └180      └190      └
DD15_HUMAN  VAQRVADEMDVMLGQEVGYSIRFEDCSSAKTILKYMTDGMLLREAMNDPLLERYGVIILDEAHERTLATDILMGVLKEVVRQRSDLKVIVMSATLDAGKF
xnu_93      VAQRVADEMDVMLGQEVGYSIRFEDCSSAKTILKYMTDGMLLREAMNDPLLERYGVIILDEAHERTLATDILMGVLKEVVRQRSDLKVIVMSATLDAGKF
seg_94      VAQRVADEMDVMLGQEVGYSIRFEDCSSAKTILKYMTDGMLLREAMNDPLLERYGVIILDEAHERTLATDILMGVLKEVVRQRSDLKVIVMSATLDAGKF
            200      └210      └220      └230      └240      └250      └260      └270      └280      └290      └
DD15_HUMAN  QIYFDNCPLLTIPGRTHPVEIFYTPEPERDYLEAAIRTVIQIHMCEEEEGDLLLFLTGQEEIDEACKRIKREVDDLGPEVGDIKIIPLYSTLPPQQQQRI
xnu_93      QIYFDNCPLLTIPGRTHPVEIFYTPEPERDYLEAAIRTVIQIHMCEEEEGDLLLFLTGQEEIDEACKRIKREVDDLGPEVGDIKIIPLYSTLXXXXXXXX
seg_94      QIYFDNCPLLTIPGRTHPVEIFYTPEPERDYLEAAIRTVIQIHMCEEEEGDLLLFLTGQEEIDEACKRIKREVDDLGPEVGDIKIIPLYSTLPPQQQQRI
            300      └310      └320      └330      └340      └350      └360      └370      └380      └390      └
DD15_HUMAN  FEPPPPKKQNGAIGRKVVVSTNIAETSLTIDGVVFVIDPGFAKQKVYNPRIRVESLLVTAISKASAQQRAGRAGRTRPGKCFRLYTEKAYKTEMQDNTYP
xnu_93      XXXXXXXXXXXXIGRKVVVSTNIAETSLTIDGVVFVIDPGFAKQKVYNPRIRVESLLVTAISKASAQQXXXXXXXXRPGKCFRLYTEKAYKTEMQDNTYP
seg_94      FEPPPPKKQNGAIGRKVVVSTNIAETSLTIDGVVFVIDPGFAKQKVYNPRIRVESLLVTAISKASXXXXXXXXXXTRPGKCFRLYTEKAYKTEMQDNTYP
            400      └410      └420      └430      └440      └450      └460      └470      └480      └490      └
DD15_HUMAN  EILRSNLGSVVLQLKKLGIDDLVHFDFMDPPAPETLMRALELLNYLAALNDDGDLTELGSMMAEFPLDPQLAKMVIASCDYNCSNEVLSITAMLSVPQCF
xnu_93      EILRSNLGSVVLQLKKLGIDDLVHFDFMDPPAPETLMRALELLNYLAALNDDGDLTELGSMMAEFPLDPQLAKMVIASCDYNCSNEVLSITAMLSVPQCF
seg_94      EILRSNLGSVVLQLKKLGIDDLVHFDFMDPPAPETLMRXXXXXXXXXXXXXXXXXTELGSMMAEFPLDPQLAKMVIASCDYNCSNEVLSITAMLSVPQCF
            500      └510      └520      └530      └540      └550      └560      └570      └580      └590      └
DD15_HUMAN  VRPTEAKKAADEAKMRFAHIDGDHLTLLNVYHAFKQNHESVQWCYDNFINYRSLMSADNVRQQLSRIMDRFNLPRRSTDFTSRDYYINIRKALVTGYFMQ
xnu_93      VRPTEAKKAADEAKMRFAHIDGDHLTLLNVYHAFKQNHESVQWCYDNFINYRSLMSADNVRQQLSRIMDRFNLPRRSTDFTSRDYYINIRKALVTGYFMQ
seg_94      VRPTEAKKAADEAKMRFAHIDGDHLTLLNVYHAFKQNHESVQWCYDNFINYRSLMSADNVRQQLSRIMDRFNLPRRSTDFTSRDYYINIRKALVTGYFMQ
            600      └610      └620      └630      └640      └650      └660      └670      └680      └690      └
DD15_HUMAN  VAHLERTGHYLTVKDNQVVQLHPSTVLDHKPEWVLYNEFVLTTKNYIRTCTDIKPEWLVKIAPQYYDMSNFPQCEAKRQLDRIIAKLQSKEYSQY~~~~~
xnu_93      VAHLERTGHYLTVKDNQVVQLHPSTVLDHKPEWVLYNEFVLTTKNYIRTCTDIKPEWLVKIAPQYYDMSNFPQCEAKRQLDRIIAKLQSKEYSQY~~~~~
seg_94      VAHLERTGHYLTVKDNQVVQLHPSTVLDHKPEWVLYNEFVLTTKNYIRTCTDIKPEWLVKIAPQYYDMSNFPQCEAKRQLDRIIAKLQSKEYSQY~~~~~
            700      └710      └720      └730      └740      └750      └760      └770      └780      └790      └
DD15_HUMAN
xnu_93
```

pos:1 col:1 DD15_HUMAN -->

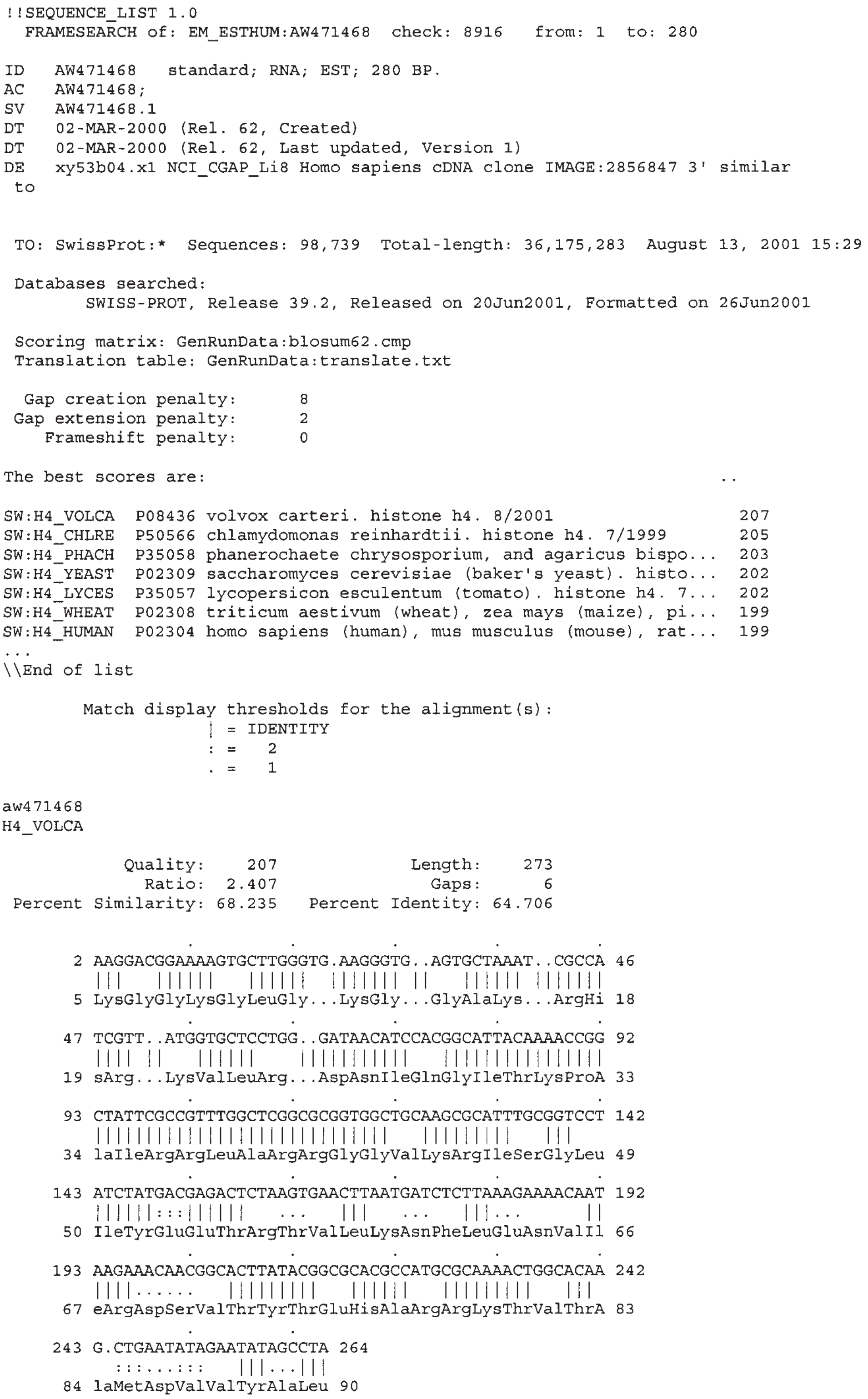
Fig. 8. *(Continued from page 478)* The SSearch/FastA output. **(B)** *(opposite page)* The list portion of the output showing the 3 significance scores. In FastA the s-w score is called the opt score. **(C)** The SeqLab editor window showing the protein sequence (single letter codes) used in this SSearch as it was retrieved from the database and after masking low complexity regions using the GCG programs Xnu (xnu_93) or Seg (seg_94). Two regions of low complexity sequence at the N-terminal portion of the protein were screened out (replaced by X's).

the Smith-Waterman algorithm for final DNA-DNA alignment be chosen when running this type of search in Fast A.

The alignments in the FrameSearch output (*see* Fig. 9A) show the alignment of codons (nucleic acid) aligned with amino acids (using the 3-letter code) with identities indicated by a line (**|**) and conservative changes indicated by a semi-colon (**:**) and a period (**.**) to indicate comparison values equal to at least +1. The problem with this type of alignment is that when the codon and amino acid residue are not identical it can be laborious to determine the identity of the amino acid in the nucleotide strand, unless of course you have memorized the entire codon table.

The histogram created by the Framesearch program appears in a separate file and looks quite different from the FastA/SSearch plots (*see* Fig. 9B). If you execute FrameSearch from the SeqLab interface, by default the output will not contain a histogram. It will only give a list of sequences sorted by the Smith-Waterman (s-w) score. However, the creation of a separate output file containing a histogram of the score distributions in the database can be selected in the options window (when run from the command line production of the histogram is on by default). The advantage of this graph is that an asterisk indicates the lowest scoring sequence in the list of returned hits and therefore allows one to rapidly determine if other sequences in the database may have significant similarity.

A



```
!!SEQUENCE_LIST 1.0
  FRAMESEARCH of: EM_ESTHUM:AW471468  check: 8916   from: 1  to: 280

ID    AW471468   standard; RNA; EST; 280 BP.
AC    AW471468;
SV    AW471468.1
DT    02-MAR-2000 (Rel. 62, Created)
DT    02-MAR-2000 (Rel. 62, Last updated, Version 1)
DE    xy53b04.x1 NCI_CGAP_Li8 Homo sapiens cDNA clone IMAGE:2856847 3' similar
 to


 TO: SwissProt:*  Sequences: 98,739  Total-length: 36,175,283  August 13, 2001 15:29

 Databases searched:
        SWISS-PROT, Release 39.2, Released on 20Jun2001, Formatted on 26Jun2001

 Scoring matrix: GenRunData:blosum62.cmp
 Translation table: GenRunData:translate.txt

  Gap creation penalty:      8
 Gap extension penalty:      2
    Frameshift penalty:      0

The best scores are:                                                  ..

SW:H4_VOLCA  P08436 volvox carteri. histone h4. 8/2001                 207
SW:H4_CHLRE  P50566 chlamydomonas reinhardtii. histone h4. 7/1999      205
SW:H4_PHACH  P35058 phanerochaete chrysosporium, and agaricus bispo...  203
SW:H4_YEAST  P02309 saccharomyces cerevisiae (baker's yeast). histo...  202
SW:H4_LYCES  P35057 lycopersicon esculentum (tomato). histone h4. 7...  202
SW:H4_WHEAT  P02308 triticum aestivum (wheat), zea mays (maize), pi...  199
SW:H4_HUMAN  P02304 homo sapiens (human), mus musculus (mouse), rat...  199
...
\\End of list

        Match display thresholds for the alignment(s):
                    | = IDENTITY
                    : =   2
                    . =   1

aw471468
H4_VOLCA

           Quality:    207             Length:    273
             Ratio:  2.407               Gaps:      6
 Percent Similarity: 68.235   Percent Identity: 64.706


                  .         .         .         .         .
       2 AAGGACGGAAAAGTGCTTGGGTG.AAGGGTG..AGTGCTAAAT..CGCCA 46
         |||   ||||||   ||||||  ||||||| ||   |||||| |||||||
       5 LysGlyGlyLysGlyLeuGly...LysGly...GlyAlaLys...ArgHi 18
                  .         .         .         .         .
      47 TCGTT..ATGGTGCTCCTGG..GATAACATCCACGGCATTACAAAACCGG 92
         |||| ||   ||||||    |||||||||||   ||||||||||||||||
      19 sArg...LysValLeuArg...AspAsnIleGlnGlyIleThrLysProA 33
                  .         .         .         .         .
      93 CTATTCGCCGTTTGGCTCGGCGCGGTGGCTGCAAGCGCATTTGCGGTCCT 142
         |||||||||||||||||||||||||||||   |||||||||   |||
      34 laIleArgArgLeuAlaArgArgGlyGlyValLysArgIleSerGlyLeu 49
                  .         .         .         .         .
     143 ATCTATGACGAGACTCTAAGTGAACTTAATGATCTCTTAAAGAAAACAAT 192
         ||||||:::||||||   ...   |||   ...   |||...      ||
      50 IleTyrGluGluThrArgThrValLeuLysAsnPheLeuGluAsnValIl 66
                  .         .         .         .         .
     193 AAGAAACAACGGCACTTATACGGCGCACGCCATGCGCAAAACTGGCACAA 242
         ||||......   |||||||||   ||||||   |||||||||   |||
      67 eArgAspSerValThrTyrThrGluHisAlaArgArgLysThrValThrA 83
                  .         .
     243 G.CTGAATATAGAATATAGCCTA 264
           :::...:::   |||...|||
      84 laMetAspValValTyrAlaLeu 90
```

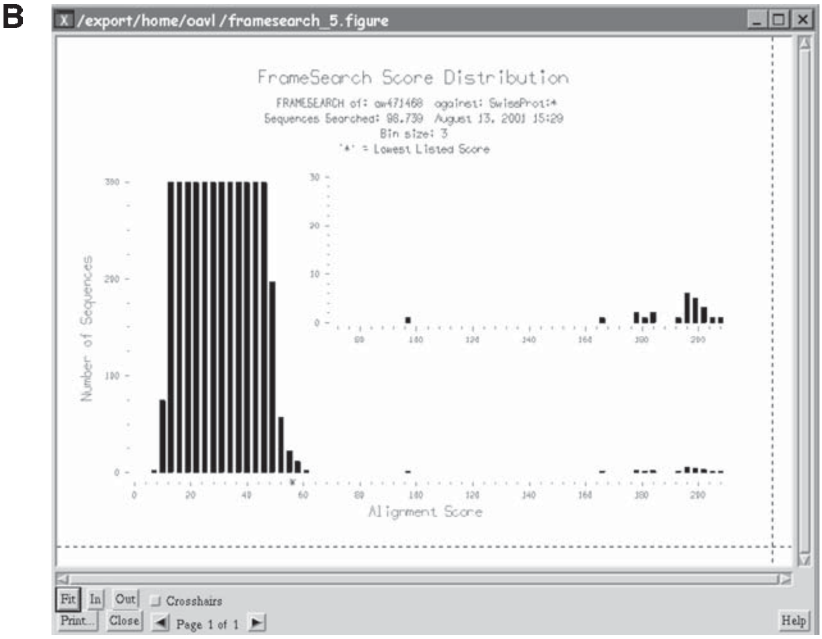

B
/export/home/oavl/framesearch_5.figure
FrameSearch Score Distribution
FRAMESEARCH of: aw471468 against: SwissProt:*
Sequences Searched: 98,739 August 13, 2001 15:29
Bin size: 3
'*' = Lowest Listed Score
Number of Sequences
300
200
100
0
30
20
10
0
0
20
40
60
80
100
120
140
160
180
200
Alignment Score
Fit
In
Out
Crosshairs
Print...
Close
Page 1 of 1
Help

Fig. 9. Results of a FrameSearch of a DNA query vs a protein database. **(A)** *(opposite page)* Note that the query is indicated as a nucleotide sequence rather than a translated protein and the amino-acid sequence of the proteins found in the database is shown using the 3-letter code to overlap with each codon. **(B)** The distribution histogram of the FrameSearch in (A). The asterisk (*) indicates the lowest score in the list of sequence hits.

Rules for Effective Database Searching

Dr. William R. Pearson outlined a series of rules for database searching. These are paraphrased with additional comments and other useful rules:

1. To determine if your sequence already exists in the databases, use the BLAST program as it is by far the fastest and most flexible search program. When searching large eukaryotic genomic DNA or EST sequences with mRNA or cDNA queries, use gapped BLAST rather than FastA or SSearch. Eukaryotic genes are made up of both exons and introns and mature mRNA only contains the exons. Only gapped BLAST will return alignments for several exons spread over large areas of genomic DNA, whereas FastA and Ssearch will find only the single best local alignment.
2. Whenever possible compare sequences at the amino acid level. If the protein sequence is known, use it to search the protein databases (FastA, SSearch). Alternatively, use a translated nucleotide vs protein search (FastX, BLASTX). Searching protein vs. protein will allow the discovery of more evolutionary related distant sequences than DNA vs DNA.
3. Search the smallest, best annotated sequence database before the large poorly annotated sequence databases. I refer to these databases as information rich vs information poor databases. Information rich databases tend to be well-curated, highly annotated, and nonredundant (i.e., Swissprot, PIR, GenBANK refseq) and have fewer partial sequence fragments, sequence errors, redundancies as

high-throughput or low quality sequence found in the whole GenEMBL, dbEST, or High Throughput Genomic (HTG) databases. In some instances it may be worthwhile to create subsets of existing databases (i.e., GeneEMBL coding sequences) using tools such as the Lookup program to improve the chance of success and performance.

4. Use statistics rather than percent similarity scores to judge the quality of a sequence match. Sometimes a sequence can be 100% identical but still insignificant if it is short, or consists of low sequence complexity. Matches with 50% similarity over regions of 20–40 amino acids occur often by chance.
5. For protein query sequences use sequence-masking programs such as Seg or Xnu to remove regions of low complexity before running your search using FastA or SSearch.
6. When evaluating the search results of sequences with a low similarity score, check that the statistical analysis is accurate by randomizing the query sequence using the GCG program Shuffle. Randomized sequences should have a lower E-value than the query sequence.
7. Try using different scoring matrices, gap opening, or extension penalties. When searching with long query sequences, changing the scoring matrix will not dramatically alter the results. However when using shorter sequences to query the database, use of more stringent matrices (Blosum 80, PAM 30) may help identify similar sequences.

Putting the Search Output to Work for You

The output files of all the database search programs described in this chapter are in a special GCG format called a *list file*. List files can be used to enter all the sequences in a list directly into the SeqLab Main or Editor windows and can be used as input to many different GCG programs. In SeqLab, when a database search result appears in the Output Manager window one simply selects the file of interest and clicks the button **Add to Main List**. Alternatively, the list can be opened from the main window by clicking the **File** pull down menu and **Open List** or used directly from the command line of different GCG programs. Now all the sequences identified by your search program are available for further GCG sequence analysis, using programs like *multiple sequence alignment* and *pairwise alignment*. An example of how to utilize this capability to full advantage is to perform a BLAST search with a sequence of interest, import the BLAST output into the GCG main window, and execute a multiple sequence alignment on the BLAST results. One can then import the file of aligned sequences into the SeqLab Editor window to view the alignment in color.

Another important but virtually unknown feature of the GCG software is that the list files (and other GCG files including rich sequence format, RSF, files) can be used to create personal databases. The program Dataset converts a group of sequences into a GCG-formatted database. The databases thus created are compatible with other GCG programs including Stringsearch, FastA/SSearch, the DatabaseBrowser, and many others. In the SeqLab interface the DataSet program appears in the menu **Functions> Utilities>Database Utilities**. To use this program, one simply selects the sequences from the main window and opens the dataset window. Once the database is created, it appears automatically in your personal SeqLab interface. Dataset can also be used from the command line and the newly created database specified at the command line for any GCG search except BLAST. Note that both Lookup and BLAST programs require databases in specific formats distinct from the normal GCG-formatted flat-file database. A database created using Dataset can be made available to everyone using the GCG software, however the systems administrator has to update certain GCG system files. The 6 files created by Dataset appear in the current working directory or can be sent to another directory using the program options or command line. Once created the files cannot be moved or renamed unless their location is edited in your *.datasetrc* file, which appears in your home directory. See the GCG Program Manual or GCG help for a more complete description of this program.

The BLAST program requires a database with a different format than a normal GCG database. GCG has provided a program similar to dataset to create BLAST databases. The program, GCGtoBLAST, is found in the same menu as dataset and works in much the same manner. However, unlike dataset, GCGtoBLAST does not automatically update the SeqLab files so that you can use your personal BLAST database. It is very likely that in order to access this database you will have to ask your systems administrator to update the file BLAST.lbds in the GCG directory (the path to this file is **../gcg/gcgcore/data/rundata/BLAST.lbds**). To install the new BLAST database called, for example, **cancer_genes_BLAST**, the full path to the database and the database name must be entered into this file in the following format **path/databasename database-type** (**protein = p, nucleotide = n**), and any comment you want. For example my database would have the following entry in the BLAST.lbds file:

/export/home/david/databases/cancer_genes_BLAST n Genes involved in cancer

This database will be visible to anyone using GCG but may not be readable by them if the Unix file permissions are set that only you can read from the files. It is probably easier to set up your own GCG-formatted databases rather than BLAST databases, but if you are undertaking very large sequence projects or only searching subsets of the GenBANK databases it may be worthwhile.

Acknowledgments

I would like to thank Dr. Jonathan Kearsey for critical reading of the manuscript; Dr. Brian Golding, Fabien Rouet, and Diego Pallares for their helpful discussions.

Glossary and Abbreviations

Algorithm A step-by-step procedure for solving a problem or accomplishing some end especially by a computer.

Bit score Used in BLAST searches the bit score represents the probability of finding the same alignment by chance. The bit score is the n in the formula 1/2n. A higher bit score indicates a better alignment. Bit scores are independent of the size of the database being searched.

Database A collection of records of similar content that can be easily accessed and managed.

Database Indexing The process of extracting different pieces of data in from flat-file database into separate files that can be searched individually and used to find the original record.

Database Records Pieces of data that are not related to one another in a structured way.

E-value The E-value score for each sequence indicates the number of sequences that would have to be searched to give a similar bit or z-score score obtained for that sequence match by random chance. A lower score is better with 0 being the best. The E-value takes into account the size of the database being searched.

Flat-file A database file containing a number of database records.

Foreign Key A piece of data in a relational database table that is the primary key of another table in the same database.

Global Alignment The detection of global similarities over the entire length of two or more sequences.

Local Alignment The detection of local or short regions of similarities in portions of two or more sequences.

Low Complexity Complexity is the probability of finding a particular amino acid at a particular position in a protein and is proportional to the composition of the whole sequence. For example in a complex protein, 10 prolines in a row are unlikely and therefore of low complexity. The same idea can apply to nucleotide sequences.

Primary Key The most important piece of data in a relational database table.

Suggested Reading

Searching by Keyword

Bailey, T. L. and Gribskov, M. (1998) Combining evidence using p-values: application to sequence homology searches, Bioinformatics 14, 48–54.

Lookup

Etzold, T. and Argos, P. (1993) SRS: an indexing and retrieval tool for flat file data libraries, Comp. Appl. Biosci. 9(1), 49–57.

Searching with Query Sequences

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1978) Altas of Protein Sequence Research and Structure, vol. 5, Suppl. 3 (Dayhoff, M. O., ed.) National Biomedical Research Foundation, Washington, DC, pp. 345–352.

Henikoff, S. and Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. U S A. 89(22), 10,915–10,919.

BLAST/NetBLAST

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool, J. Mol. Biol. 215, 403–410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25, 3389–3402.

Shpaer, E. G., Robinson, M., Yee, D., Candlin, J. D., Mines, R., and Hunkapiller, T. (1996) Sensitivity and selectivity in protein similarity searches: a comparison of Smith-Waterman in hardware to BLAST and FastA, Genomics 38, 179–191.

Wootton, J. C. and Federhen, S. (1996) Analysis of compositionally biased regions in sequence databases, Methods Enzymol. 266, 554–571.

FastA Suite/Ssearch/FrameSearch

Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence comparison, Proc. Natl. Acad. Sci. USA, 85(8), 2444–2448.

Pearson, W. R. (1995) Comparison of methods for searching protein sequence databases, Protein Sci. 4, 1145–1160.

Smith, T. F. and Waterman, M. S. (1981) Comparison of Bio-Sequences, Adv. Appl. Math. 2, 482–489.

Rules for Effective Database Searching

Pearson, W. R. (1996) Effective protein sequence comparison, Methods Enzymol. 266, 227–258.

Pearson, W. R. (2000) Flexible sequence similarity searching with the FastA3 program package, Methods Mol. Biol. 132, 185–219.

C. Identifying Functional and Structural Sequence Elements

29 Pattern Discovery

Methods and Software

Broňa Brejová, Tomáš Vinar, and Ming Li

Introduction

A pattern is a feature that occurs repeatedly in biological sequences, typically more often than expected at random. Patterns often correspond to functionally or structurally important elements in proteins and DNA sequences. Pattern discovery is one of the fundamental problems in bioinformatics. It has applications in multiple sequence alignment, protein structure and function prediction, drug target discovery, characterization of protein families, and promoter signal detection.

Regions important to the structure or function of the molecule tend to evolve more slowly. In particular an occurrence of a conserved motif in a protein may imply that this region may be involved in the interaction with some other protein, may comprise the active site of an enzyme, or may be important for the tertiary structure of the protein. Attempts have been made to organize proteins and protein domains from different organisms into families based on their evolutionary relations, and structural and functional similarity. Sequences in one family often share one or several common motifs and these motifs are used to characterize the family. Several databases containing motifs characterizing protein families have been established. Newly discovered proteins can be assigned to a family by searching a database of motifs. We may then associate the function or structure to the new protein based on the knowledge we have about the other members of the family.

Nucleotide sequences outside of protein coding regions tend to be less conserved, except where they are important for function, that is, where they are involved in the regulation of gene expression. Some regulatory elements are located in promoter regions upstream from genes. Identifying promoters in genomic sequences is difficult, especially in eukaryotic genomes because they do not have a common core promoter, but rather consist of multiple regulatory factors distributed over long distances. Adding to the complexity is low number of availab le annotated promoters. The best programs are only able to identify about half of eukaryotic promoters.

If a pattern characteristic for a binding site of a certain transcription factor is known, we can find occurrences of this pattern in promoter regions of known genes. This helps us to understand how these genes are regulated, under which conditions they are transcribed, and may even help to infer a function of a gene.

New binding motifs of transcription factors can be discovered by considering upstream regions of co-regulated genes and identifying motifs that occur in these regions more frequently than elsewhere. For example groups of co-regulated genes can be identified by analyzing gene expression data. Provided that a transcription factor is conserved between two species, we may discover its binding sites by identifying conserved sequences in promoter regions of these two genomes.

Related molecules, usually proteins or RNA, sometimes do not display significant similarity at the sequence level. However, significant similarity can be found in their secondary or tertiary structure. Discovery of structural motifs in proteins and RNA molecules has also been studied, but this work is beyond the scope of this chapter.

The availability of several fully sequenced genomes has enabled scientists to identify homologies shared between two genomes. Such conserved regions are likely to correspond to functionally important elements. This information has been applied to predict genes, discover new regulatory elements, and reveal evolutionary relationships between species and types of evolutionary changes to genome organization. Identifying possibly long similarities between two long sequences can be considered a special case of pattern discovery, yet the large amounts of data require special consideration from the computer science point of view.

In this chapter, we introduce the basics of algorithms used to discover patterns. We also discuss the goals of such algorithms and how to statistically verify their results. Many programs for pattern discovery and databases of biologically relevant patterns are available. We provide an overview and links to these important tools in a supplement located in the accompanying CD-ROM.

Pattern Discovery

What is a Pattern

A pattern is an element that has multiple occurrences in a given set of biological sequences. In this section we outline how to represent a pattern. Probably the simplest representation of a pattern is a list of its occurrences in the given sequences which is sufficient to specify the pattern. However, it is not convenient for further use as it is difficult to decide whether or not a new sequence contains occurrences of the pattern. Therefore we usually represent a pattern by describing properties shared by all its occurrences. Such representation is more succinct and allows easier searches for new occurrences.

Deterministic Patterns

The simplest kind of a pattern is a *consensus sequence*. For example *TATAAAA* is the TATA box consensus sequence. Whenever we find string *TATAAAA* we say it is an occurrence of this pattern. Of course not all such occurrences correspond to a real TATA box and not every TATA box perfectly matches the consensus. The latter of these two problems can be solved by allowing a certain degree of flexibility in the pattern. This can be achieved by adding some of the following frequently used features.

Let Σ be the alphabet of all possible characters occurring in the sequences (i.e., $\Sigma = \{A,C,G,T\}$ for DNA sequences and Σ is a set of all 20 amino acids for protein sequences).

- An ambiguous character matches any character from a subset of Σ. Such a subset is denoted by a list of its elements enclosed in square brackets. For example, *[LF]* is a set containing L and F. *A-[LF]-G* is a pattern in a notation used in PROSITE database. This pattern matches 3-character subsequences starting with A, ending with G and having either L or F in the middle.

For nucleotide sequences there is a special letter for each set of nucleotides, where:

R = [AG], Y = [CT], W = [AT], S = [GC], M = [AC], K = [GT], B = [CGT], D = [AGT], H = [ACT], V = [ACG], N = [ACGT]

- *Wildcard* or *don't care* is a special kind of ambiguous character that matches any character from Σ. Wildcards are denoted *N* in nucleotide sequences, *X* in protein sequences. They may also be denoted by a dot '.'. A group of one or several consecutive wildcards is called a *gap* and patterns allowing wildcards are often called *gapped patterns*.
- *Flexible gap* is a gap of variable length. In the PROSITE database it is denoted by *x (i,j)* where *i* is the lower bound on the gap length and *j* is an upper bound. Thus *x*(4,6) matches any gap with length 4, 5, or 6. Fixed gap of length *i* is denoted *x* (*i*) (e.g., *x*(3) = *xxx*). Finally * denotes a gap of any length (possibly 0).

String *F*-*x*(5)-*[LF]*-*x*(2,4)-*G*-*-H is an example of a PROSITE pattern containing all of the above features.

Patterns with Mismatches

One can further extend the expressive power of deterministic patterns by allowing a certain number of mismatches. The most commonly used type of mismatches are substitutions. In this case subsequence *S* matches pattern *P* with at most *k* mismatches, if there is a sequence *S*' exactly matching *P* that differs from *S* in at most *k* positions. Sometimes we may also allow insertions or deletions, i.e., the number of mismatches would be an edit distance between the substring *S* and a closest string matching the pattern *P*.

Position Weight Matrices

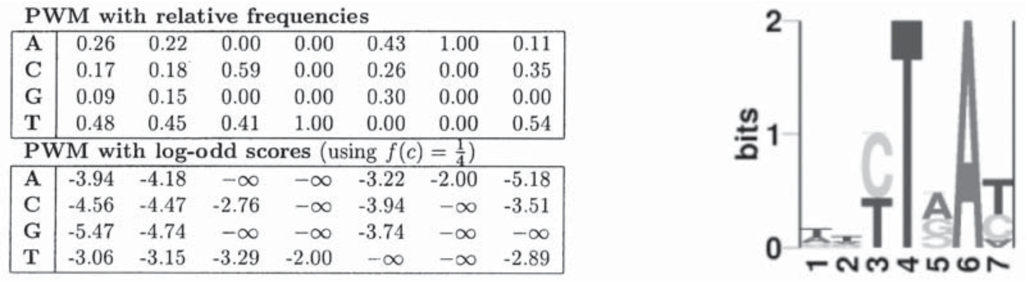
So far we have explored only deterministic patterns. A deterministic pattern either matches the given string or not. However, even the most complicated deterministic patterns cannot capture some subtle information hidden in a pattern. Let us assume we have a pattern. The first position is *C* in 40% cases and *G* in 60% cases. The ambiguous symbol [*CG*] gives the same importance to both nucleotides. It does not matter in strong patterns, but it may be important in weak patterns, where we need to use every piece of information to distinguish the pattern from a random sequence.

Probabilistic patterns are probabilistic models that assign a probability to each sequence that was generated by the model. The higher the probability, the better the match between the sequence and the pattern.

The simplest type of probabilistic pattern is a *position-weight matrix* (PWM). PWMs are also sometimes called a *position-specific score matrix* (PSSM), or a *profile* (however, term *profile* is also used for more complicated patterns allowing gaps). PWM is a simple ungapped pattern specified by a table. This table shows the relative frequency of each character at each position of the pattern (*see* Fig. 1 for an example).

Assume that the pattern (i.e., PWM) has length *k* (number of columns of the table). The score of a sequence segment $x_1 \ldots x_k$ of length *k* is

$$\prod_{i=1}^{k} \frac{A[x_i,i]}{f(x_i)}$$



PWM with relative frequencies

A	0.26	0.22	0.00	0.00	0.43	1.00	0.11
C	0.17	0.18	0.59	0.00	0.26	0.00	0.35
G	0.09	0.15	0.00	0.00	0.30	0.00	0.00
T	0.48	0.45	0.41	1.00	0.00	0.00	0.54

PWM with log-odd scores (using $f(c) = \frac{1}{4}$)

A	-3.94	-4.18	$-\infty$	$-\infty$	-3.22	-2.00	-5.18
C	-4.56	-4.47	-2.76	$-\infty$	-3.94	$-\infty$	-3.51
G	-5.47	-4.74	$-\infty$	$-\infty$	-3.74	$-\infty$	$-\infty$
T	-3.06	-3.15	-3.29	-2.00	$-\infty$	$-\infty$	-2.89

Fig. 1. Position weight matrix of vertebrate branch point in the form of a table and corresponding visual representation as a sequence logo. The sequence logo was created using RNA Structure Logo, an on-line tool available at Website: http://www.cbs.dtu.dk/gorodkin/appl/slogo.html.

where $A\ [c,i]$ is the entry of position weight matrix corresponding to position i of the pattern and character c, and $f(c)$ is the background frequency of character c in all sequences considered. This product represents the odd-score that the sequence segment $x_1 \ldots x_k$ belongs to the probability distribution represented by the PWM. In order to simplify the computation of the score we can store log-odd scores $A'[c,i] = \log_2 A[c,i]/f(c)$ in a table, in place of using frequencies $A[c,i]$. Then the following formula yields the log-odd score:

$$\sum_{i=1}^{k} A'[x_i,i]$$

Position-weight matrices can be visualized in the form of sequence logos (*see* Fig. 1). Each column of a sequence logo corresponds to one position of the pattern. Relative heights of the characters in one column are proportional to the frequencies $A[c,i]$ at the corresponding position of the pattern. At each position, the characters are ordered according to their frequency, with the most frequent character on top. Each column is scaled so that its total height is proportional to the information content of the position, computed as:

$$\log_2 |\Sigma| + \sum_{c} A[c,i] \log_2 A[c,i]$$

Note: $\log_2 |\Sigma|$ is added to obtain positive values. It is dependent on the size of alphabet Σ. Sequence logos have been developed further to consider background distribution and to invert characters that occur less frequently than expected.

An examination of a sequence logo reveals the most conserved positions (highest columns) and consensus characters at all positions (highest letter in the column). The size of characters in different columns cannot be directly compared.

Stochastic Models

The patterns discussed thus far are explicit in the sense that the user can easily see important characteristics of the occurrences of a pattern. Sometimes it is advantageous to represent a pattern in a more implicit form, usually as a discrimination rule, to decide whether a given sequence is an occurrence of the modeled pattern or not. It can be based on a stochastic model, such as *Hidden Markov Model* (HMM), or can employ machine learning methods, such as neural networks.

It is questionable whether such rules constitute a pattern at all. Obviously, they can be trained (pattern discovery) and then they can be used for discrimination (pattern matching). Therefore, they are applicable in pattern-related tasks such as protein family classification and binding site discovery. In some cases (such as HMMs with simple topology) it is even possible to obtain some information about the pattern modeled. For example, the relative frequencies of characters at individual conserved positions.

Pattern Discovery vs Pattern Matching

There are two fundamentally different tasks related to identifying new patterns in biological sequences. The first one is called *pattern matching*. This finds new occurrences of a known pattern. Many consensus sequences are known in biology and it is important to have tools that will allow one to find occurrences of known patterns in new sequences.

There are specialized software tools for pattern matching. Some are quite general, i.e., they allow the user to specify a pattern as part of the input in a specific form. Others are built to recognize only one pattern. Many specialized tools are available for recognizing splicing signals, different regulatory elements, and special structural elements. Authors of such specialized tools fine-tune the parameters of the system to increase the accuracy of the prediction.

Although pattern matching is very important, we will concentrate on a different kind of pattern-related problem, called *pattern discovery*. The task is to identify a new pattern in a set of several sequences.

The input for pattern discovery programs consists of several sequences, expected to contain the pattern. Input sequences are typically related in some way, e.g., they are members of the same protein family, functionally related sequences, or upstream regions of co-regulated genes.

Goals of Pattern Discovery

The goal of pattern discovery is to identify an unknown pattern in a given set of sequences. There are a great number of potential patterns and it is often difficult to decide which of them are the most promising. Defining the *best* pattern depends on the intended use of the pattern. We will consider two possible scenarios: 1) classification, i.e., we want to characterize members of some sequence family and distinguish them from nonmembers, and 2) identification of significant patterns, i.e., we want to discover patterns that are unlikely to occur by chance and would therefore probably have functional or structural significance.

Classification

In a classification scenario we want to identify motifs that best characterize a given protein family. The motifs thus identified are then used as classifiers. For example, given an unknown protein we can classify it as a member or nonmember of a family, based on whether it contains the motifs characteristic for that family. This is a typical machine learning problem: given a set of sequences belonging to the family (positive examples) and a set of sequences not belonging to the family (negative examples), find a function *f*, which decides for each protein whether or not it belongs to the family. In the context of motif discovery we consider classes of functions *f*, that match some discovered patterns against the unknown sequence. Note, that negative examples are simply other known proteins taken from protein databases such as SWISS-PROT.

The common strategy in pattern discovery is to use only positive examples. The most significant motif in the family is found in a hope that it will not occur elsewhere. Negative examples are only used to evaluate the prediction. Thus, the task is converted to the second scenario, described in the following.

Identifying Significant Patterns

Motif discovery is not always formulated as a classification problem. For example let's identify a regulatory element in a set of regions likely to contain this element. However, it does not mean that this element cannot occur in other places in the genome or that all of these sequences must contain a common regulatory element. We are also interested in identifying conserved regions (protein family motifs) that may indicate structurally or functionally important elements, regardless of whether they have enough specificity to distinguish this family from other families. In this context it is more complicated to precisely formulate the question.

The usual approach is to find the *highest scoring* pattern within a well-defined class of patterns (e.g., PROSITE patterns, *see* Subheading "What is a Pattern?"), that has sufficient *support*. Various approaches use different scoring functions and support measures and consider different classes of patterns. Support of a pattern is usually the number of sequences in which the pattern occurs. We can require that the pattern should occur in all sequences or there is a minimum number of occurrences specified by the user. In some cases the number of occurrences is not specified but it is a part of the scoring function. A longer pattern with fewer occurrences is sometimes more interesting than a shorter pattern with more occurrences. The situation is more complicated in the case of probabilistic patterns, such as HMMs. Deterministic patterns either match the sequence or not (zero or one), whereas probabilistic models give a probability between 0 and 1. Therefore, there are different degrees of *matching*. It is necessary to set some threshold on what should be considered a match or to integrate these matching probabilities to the scoring scheme.

Methods for scoring patterns also differ. A score can reflect the pattern itself only (e.g., its length and degree of ambiguity), or it can be based on the occurrences of the pattern (their number, how much these occurrences differ from the pattern). Scoring functions are sometimes based on statistical significance. For example, we may ask: what is the probability that the pattern would have so many occurrences if the sequences were generated at random? If this probability is small, the pattern is statistically significant. (For a more detailed discussion of statistical significance of patterns, *see* "Assessment of Pattern Quality" in this chapter.) The goal of an algorithm is to find the highest scoring pattern, or to find several best scoring patterns, or all patterns with some predefined level of support and score.

Algorithms for Pattern Discovery

Exhaustive Search

Many computer science problems related to pattern discovery are computationally hard tasks. One cannot hope to find a fast algorithm that would guarantee the best possible solution. Thus, many approaches are based on the exhaustive search. Although such algorithms in the worst case may run in exponential time, they often use sophisticated pruning techniques that make the search feasible.

Enumerating All Patterns

The simplest exhaustive search works as follows. All possible patterns satisfying constraints given by the user are enumerated. For each pattern the program finds each occurrence in the input sequences and based on these occurrences assigns a score or statistical significance to the pattern. We can then output patterns with the highest score or all patterns with scores above some threshold. For example, if we want to identify the most significant nucleotide pattern of length 10 with at most 2 mismatches, we can enumerate all possible strings of length 10 over the alphabet $\{A,C,G,T\}$ (there are $4^{10} = 1{,}048{,}576$ such strings). Each string is a potential pattern. We find all occurrences with at most 2 mismatches in the input sequences and compute the score. We report the pattern with the highest score.

This method is suitable only for short and simple patterns, because the running time increases exponentially with the length of the pattern. The number of possibilities is even larger if we allow patterns to contain wildcards, ambiguous characters, and gaps. On the other hand, the advantage of this method is that with increasing length of the input sequences the running time usually increases linearly. Therefore the enumeration approach is suitable for identifying short patterns in large data sets.

An exhaustive search is guaranteed to identify the best pattern. We may easily output an arbitrary number of high scoring patterns, we may also choose relatively complicated scoring functions, providing that they can be easily computed based on the pattern and its occurrences. We can also allow mismatches, even insertions and deletions.

Application of Enumerative Methods

Many protein binding sites in DNA are actually short ungapped motifs, with certain variability. They can be quite well-modeled with simple patterns allowing a small number of mismatches. Therefore we can apply exhaustive searches to identify these types of binding sites. Simplicity of the exhaustive search allows one to develop sophisticated methods for pattern statistical significance estimation.

Enumerating Gapped Patterns

In some contexts it is more reasonable to search for patterns with gaps. MOTIF is an example of such system. MOTIF finds patterns with 3 conserved amino acids separated by two fixed gaps (for example $A\ldots Q\ldots.I$). The gaps can have length $0,1,\ldots,d$ where d is a parameter specified by the user. The number of possible patterns is 20^3d^2. MOTIF does not allow any mismatches, but the pattern does not need to occur in all sequences.

If the sequences contain a conserved region of more than 3 positions, there will be many patterns, each containing a different subset of the conserved positions from this region. Therefore, in the following step the algorithm removes the patterns occurring close to each other. For each of the remaining patterns all occurrences of the pattern are aligned. Based on the alignment the pattern is extended by finding a consensus in the columns of the alignment. Patterns are also extended to both sides where possible.

Pruning Pattern Enumeration

If we want to identify longer or more ambiguous patterns, we cannot use a straightforward exhaustive search. For example, assume that we want to identify a long ungapped pattern occurring (possibly with some mismatches) in at least k sequences. We will start from short patterns (for example patterns of length 1) that appear in at

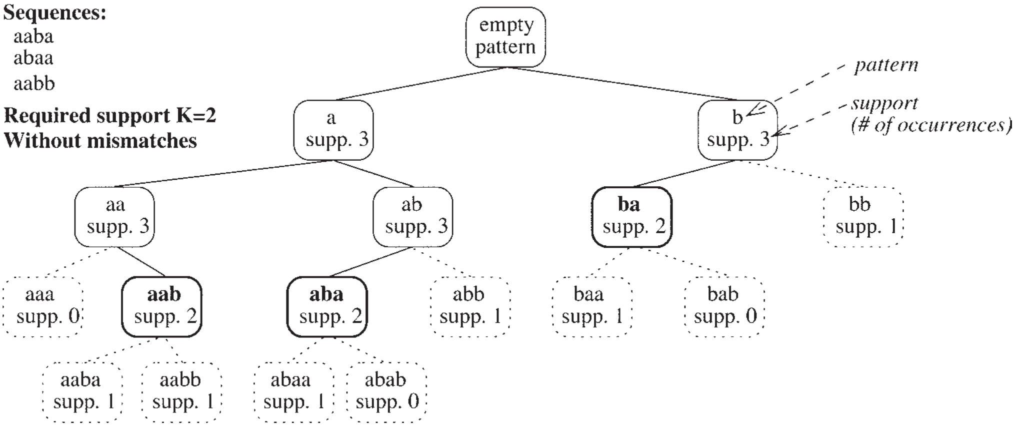


Fig. 2. One way to improve an exhaustive search is to search in a tree of all possible patterns. When we discover a node corresponding to a pattern that does not have enough support, we do not continue to search its subtrees. Dashed nodes do not have enough support. Bold nodes are patterns that cannot be further extended.

least *k* sequences and extend them while the support does not go below *k*. In each step we extend the pattern in all possible ways and check whether the new patterns still occur in at least *k* sequences. Once we obtain a pattern that cannot be extended without loss of support, this pattern is maximal and can be written to the output. This search strategy is actually a depth first search of a tree of all possible sequences (*see* Fig. 2). We prune branches that cannot yield supported patterns.

Improvements of this kind perform well in practice. However, in the worst-case scenario the running time can still be exponential. The main advantage of such improvements is that they allow a search for longer and more complicated patterns than a simple exhaustive search. Examples of this strategy include the Pratt algorithm described in detail below and the first scanning phase of TEIRESIAS algorithm.

Pratt

Pratt is an advanced algorithm based on applying the depth first search to a tree of patterns. Pratt discovers patterns containing flexible gaps (with a lower and upper bound of the gap length) and ambiguous symbols. Each pattern discovered is required to exactly match at least some predetermined number of sequences. The user has to specify several parameters that restrict the type of patterns. These include the maximum total length of the pattern, maximum number of gaps, maximum number of flexible gaps, and set of allowed ambiguous symbols.

To reduce the size of the output and the running time, the program does not report patterns that are less specific than other discovered patterns. Here pattern *A* is more specific than pattern *B*, if any sequence that matches *A* must also match *B* (for example, *B* is less specific if it can be obtained from *A* by replacing nonambiguous characters with ambiguous characters, or making a gap more flexible). This is achieved by a special scoring function that gives a higher score to more specific patterns.

In each step of the depth first search, we take an existing pattern with sufficient support and we add a gap (possibly of length 0) and another character or ambiguous character. All such possibilities are tried. New patterns without sufficient support are then discarded. This uses a special data structure that makes the search faster. Additional optimizations are carried out by discarding patterns that cannot be extended to the most significant pattern.

The Pratt algorithm is guaranteed to find the pattern with the highest score when no flexible gaps are allowed. If we allow flexible gaps, the returned pattern may not be the highest scoring since a heuristic is used to speed up the search.

Exhaustive Search on Graphs

Not all exhaustive search methods enumerate all relevant patterns. It is also possible to enumerate all combinations of substrings of given sequences that can be possible occurrences of a pattern. Assume we have n sequences and we want to identify a pattern of given length L, that occurs in all sequences with at most d mismatches. Then any two occurrences of such a pattern differ at most in $2d$ positions, because they both differ from the pattern in at most d positions. Therefore, we can identify the pattern by finding a group of n substrings of length L, each from a different sequence, such that any two substrings differ in at most $2d$ positions.

This can be formulated as a problem in graph theory as follows. Each substring of length L will be a vertex of a graph G. Vertices corresponding to two substrings will be connected by an edge if the substrings are taken from different sequences and differ in at most $2d$ positions (*see* Fig. 3A). This graph is n-partite. This means that it can be partitioned to n partitions so that there is no edge between vertices in the same partition. In this case partitions correspond to individual sequences. We want to find a set of n vertices such that any two vertices are connected by an edge. Such a set of vertices is called a *clique*.

The problem of finding a clique is known to be NP-hard. NP-hard problems are computationally difficult problems. There is no known algorithm for solving an NP-hard problem in polynomial time. One way to find a clique is to enumerate combinations of vertices and test each combination for being a clique. Such an approach has an exponential running time because there are many possible combinations. In order to make the software practical we need to add careful pruning. This eliminates large groups of vertex combinations that are guaranteed not to contain a clique.

The WINNOWER algorithm eliminates combinations by first modifying the graph itself. It attempts to reduce the number of edges in the graph, removing only edges that cannot be part of any clique of size n. In this way we may obtain a graph with less edges that will be easier to search for a clique.

Even if we find a combination of n substrings of length L, each pair differing in at most $2d$ positions, it does not guarantee that we have found a pattern. For example assume that we want a pattern of length $L = 4$ with at most one mismatch and we have found the following 3 occurrences: AAAA, BBAA and CCAA. Any two occurrences differ in exactly $2 = 2d$ positions, but there is no pattern that would differ from each occurrence in at most one position. However, we may assume that this would not happen very often and that most combinations found will actually correspond to a pattern.

Usually the user wants to know the set of occurrences and the corresponding pattern. One way to identify the pattern is to enumerate all patterns that occur within distance d from one chosen occurrence. There are at most $\binom{L}{d}$ $(|\Sigma| - 1)^d$ such patterns. This number is exponential in d but not in L and d is typically small. For each possible pattern we verify whether it is within distance d from all other occurrences as well.

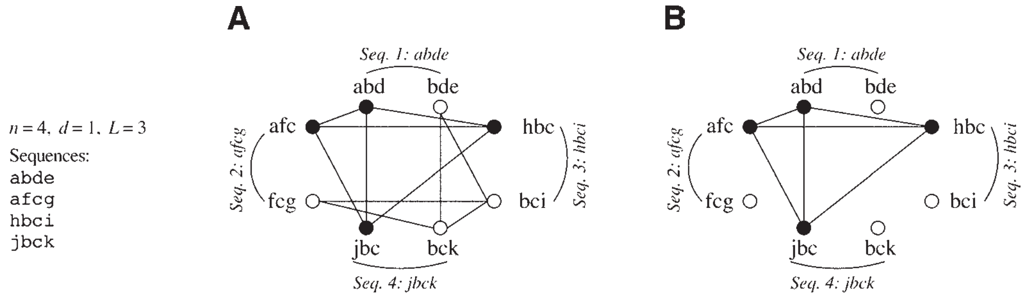


Fig. 3. **(A)** shows the graph corresponding to the depicted set of sequences. **(B)** shows the same graph after removing edges with the WINNOWER algorithm. This graph contains exactly one clique corresponding to pattern abc.

The search can be further pruned by using knowledge about the other occurrences. Alternatively we can use the set of occurrences as a starting point for Gibbs sampling or other iterative methods (*see* "Gibbs Sampling" and "Other Iterative Methods"). This is of course not guaranteed to find the pattern with specified parameters, even if one exists.

Creating Long Patterns from Short Patterns

A pattern cannot be significant unless it is sufficiently long. However long patterns are more difficult to identify using enumerative techniques. One possible approach to identifying long patterns is to start with shorter patterns and then combine them together. Perhaps the most elegant example of such an algorithm is TEIRESIAS. This algorithm is based on a well-organized exhaustive search through possible combinations of shorter patterns. In the worst case the algorithm runs in exponential time, but in practice it works very well. Further effort has yielded a different algorithm that runs in polynomial time (*see* "Improvement of Running Time").

TEIRESIAS Algorithm

TEIRESIAS searches for so-called (L,W) patterns (L and W are constants specified by the user). Pattern P is an (L,W) pattern if it meets the following criteria:

- P consists of characters from Σ and wildcards '.'
- P starts and ends with a character from Σ (i.e., non-wildcard)
- Any substring of P that starts and ends with a non-wildcard and contains exactly L non-wildcards has a length W at most (this condition is called *density constraint*).

The density constraint eliminates patterns with long stretches of wildcards. Consider for example $L = 3$ and $W = 5$. String $AF..CH..E$ is a valid (3,5) pattern, however string $AF.C.H..E$ is not (substring $C.H..E$ has length 6).

Maximal Patterns

TEIRESIAS discovers all (L,W) patterns that occur in at least K input sequences (K 2 is also specified by the user). However, out of several patterns having the same set of occurrences it outputs only one pattern. This is selected as follows.

Pattern P is said to be more specific than pattern Q if Q can be derived from P by removing several (possibly 0) characters from both ends of P and replacing several (possibly 0) non-wildcards with wildcards. For example $AB.CD.E$ is more specific than $AB..D$.

If pattern P is more specific than pattern Q, then every occurrence of P is also an occurrence of Q. If Q has the same number of occurrences as P, it is not useful to report both P and Q because they have the same set of occurrences and P contains more information. Therefore the algorithm only outputs pattern P if a more specific pattern with the same number of occurrences does not exist. Patterns reported by the algorithm are called *maximal*.

Note that if P is more specific than Q and Q has more occurrences than P, i.e., Q has greater support, Q is outputted as well. Although Q has a smaller specificity, it has greater support.

Algorithm

The TEIRESIAS algorithm is based on the concept that if a pattern P is a (L,W) pattern occurring in at least K sequences, then its subpatterns are also (L,W) patterns occurring in at least K sequences. Therefore the algorithm assembles the maximal patterns from smaller subpatterns.

TEIRESIAS works in two phases. In the first phase (called *scanning phase*), it finds all (L,W) patterns occurring in at least K sequences that contain exactly L non-wildcards. This is carried out by a pruned exhaustive search (*see* "Enumerating All Patterns"). In the second, *convolution phase* these elementary patterns are extended by gluing them together. In order to determine whether two patterns P and Q can be glued together we compare the suffix of P containing exactly $L – 1$ non-wildcards to the prefix of Q containing exactly $L – 1$ non-wildcards. If the suffix and the prefix are equal, P and Q can be glued together so that the $L – 1$ non-wildcards overlap. The list of occurrences of the resulting pattern can be constructed from the lists of occurrences of P and Q (we do not need to scan all sequences). Only when the resulting pattern occurs at least K times, is it retained.

For example let $P = AB.CD.E$ and $Q = DFE.G$ (with $L = 3$, $W = 5$). In this case P and Q cannot be glued together, because $D.E \quad DF$. However if $Q = D.E.G$ we can glue them together obtaining $AB.CD.E.G$. If occurrences of P are (1,1), (2,3), (2,6), (4,7) (each pair gives a sequence and a position in the sequence) and occurrences of Q are (1,5), (2,8), (2,10), then the list of occurrences for the new pattern is (1,1), (2,6).

In the convolution phase we take each elementary pattern, and we try to extend it on both sides by gluing it with other elementary patterns in all possible ways (depth first search). Any pattern that cannot be extended without loss of support can potentially be maximal. However we can still obtain non-maximal patterns in the output and some patterns can be generated more than once. Therefore a list of patterns written to the output is maintained. In this manner we can check any newly generated pattern with the list and if the list contains a more specific pattern with the same set of occurrences we simply discard the new pattern.

The TEIRESIAS algorithm is an exact algorithm. It is guaranteed to find all (L,W) maximal patterns supported by at least K sequences. The number of such patterns can be exponential. In such cases TEIRESIAS will require exponential time to complete. However, such a situation is not likely to occur with real data. For example, the entire GenPept database with 120 million amino acids contains only 27 million maximal patterns. Experimental studies suggest that the running time of TEIRESIAS algorithm is linear in the number of patterns it outputs.

Patterns discovered by the TEIRESIAS algorithm are not very flexible. First of all, the only mismatches allowed are wildcard characters. Newer versions of TEIRESIAS can now also identify patterns containing ambiguous characters representing prespecified groups of characters from Σ. Second, TEIRESIAS patterns do not allow gaps with flexible length. This problem can be addressed by the post-processing phase, where the patterns found are combined into larger patterns separated by flexible gaps. However, such methods do not guarantee that all patterns of the specified form will be found.

Improvement of Running Time

Irredundant Patterns

One of the drawbacks of TEIRESIAS is the potential exponential size of the output and thus exponential running time. A new algorithm has recently been developed that computes only a subset of maximal patterns, called *irredundant patterns*. Any maximal pattern can be easily obtained from the set of irredundant patterns. The main advantage of the new method is that in any input of length n there are at most $3n$ irredundant patterns and these patterns can be found in $O(n^3 \log n)$ time. This is a substantial theoretical improvement compared to traditional exponential algorithms. However neither implementation of this algorithm, nor experimental study demonstrating the application of this approach is available to date.

Iterative Heuristic Methods

So far we have considered algorithms guaranteed to identify the best pattern. However for more complicated types of patterns we cannot hope to do so. We have to use heuristic approaches that may not find the best pattern, but may converge to a local maximum. The most important example of such technique is Gibbs sampling.

Gibbs Sampling

In its simplest version (the Gibbs sampling method), we are looking for the best conserved ungapped pattern of fixed length W in the form of a position weight matrix. We assume that the pattern occurs in all sequences.

The algorithm is carried out in iterations. The result of each iteration is a set of subsequences of length W: one from each sequence. This set of subsequences represents the occurrences of the pattern. We can compute a position weight matrix characterizing the pattern from this set of occurrences. The algorithm works as follows:

- Randomly select one subsequence of length W from each input sequence. These subsequences will form our initial set of occurrences. Denote o_i occurrence in sequence i.
- Iteration step.
- Randomly select one sequence i.
- Compute the position weight matrix based on all occurrences except o_i. Denote this position weight matrix P.
- Take each subsequence of sequence i of length W and compute a score of this subsequence according to matrix P.
- Select a new occurrence o_i' randomly among all subsequences of i of length W using the probability distribution defined by the scores (a higher score means higher probability).

- Replace o_i with o_i' in the set of occurrences.
- Repeat iteration, until a stop condition is met.

The Gibbs sampling algorithm does not guarantee that the position weight matrix and set of occurrences giving the best score will be found. Instead, the algorithm can converge to a local maximum. The method is fast, making it suitable for many applications.

Several problems related to Gibbs sampling have been identified and addressed in subsequent work.

- **Phase shifts.** Assume that the optimal set contains occurrences starting at positions 8, 14, 22, and 7 of the corresponding sequences. If we start with position 21 in the third sequence, the whole system is likely to converge to the set of occurrences 7, 13, 21, and 6 instead.

  The problem was addressed by introducing an additional randomized step. In this step, scores of the occurrences shifted by several characters are computed. One random shift is selected with a probability distribution corresponding to the scores. Authors of PROBE reduce or extend patterns on both sides in a similar manner.
- **Multiple patterns.** Sometimes it is appropriate to define a pattern as a sequence of several consecutive subsequences of fixed length separated by variable length gaps. In this case each occurrence is represented by several short subsequences in the sequence rather than one. It is possible to identify such patterns using a modified Gibbs sampling approach that employs dynamic programming in the process of ranking and choosing a new candidate occurrence. Lengths of subsequences and their number are specified beforehand.
- **Pattern width.** We have assumed, that the pattern width is fixed and is specified by the user. Most of the time it is not a reasonable assumption, especially if we are looking for multiple patterns separated by variable length gaps.

  In PROBE a genetic algorithm is used to determine the parameters of patterns (i.e., the number of subsequences and their lengths). Two sets of parameters can be recombined (take part of the first and part of the second set) and in this manner a better set of parameters may be obtained. Sets of parameters for recombination are chosen at random with a distribution proportional to their score (called *fitness*). Fitness of the set of parameters is determined by the Gibbs sampling procedure.
- **Gapped patterns.** Not all positions within a continuous block of length W are necessarily important for the function of this block. Rather we want to create a pattern which is gapped, i.e., only $J < W$ positions are used to form the model. This problem can be addressed by introducing yet another randomized step, in which we replace one of the J positions included in the pattern by one of the $W - J + 1$ positions, which are not included in the pattern. The choice is again random with a distribution of probabilities proportional to the corresponding scores.

Other Iterative Methods

Several other approaches use iterative methods similar to Gibbs sampling. Typically the algorithm starts with some pattern and finds the best fitting occurrence of this pattern in each sequence. Based on these occurrences it builds a pattern that best matches the occurrences. This process is then repeated with the new pattern until no improvement is obtained. The main difference between this algorithm and Gibbs sampling is that all sequences are used to define the new pattern and subsequently the position of the new pattern is refined in all sequences. The process is completely deterministic, and of course has no guarantee to find the global optimum.

This strategy was used to identify ungapped deterministic patterns of a given length that matches all sequences with mismatches. The goal is to minimize the total number of mismatches. By using different methods a set of occurrences of some unknown candidate pattern is obtained that can be refined by an iterative method. In each step a new pattern is computed by taking the most frequent character in each position (based on the frequencies in the occurrences). The method is further improved to remove nonsignificant columns from consideration to obtain a gapped pattern.

An iterative method has also been used to detect coiled coil regions in histidine kinase receptors. Coiled coils were previously detected in other protein families; therefore, the statistical properties of such regions are known, although they may be somewhat different for this family. In this example the goal was to identify the distribution of residues and pairs of residues at different distances in a sliding window of fixed length, provided that the window is from a coiled coil region. The process started with taking the known distribution from other families. Based on this information each position of the sliding window was scored and the best scoring positions were the candidates for coiled coil regions. A random sample of these candidates was used to compute a new distribution. This process was iterated. In each step a pseudocount from the known distribution of other families was added. In contrast to the previous method, this is randomized, and due to pseudocounts the result cannot extensively diverge from the original pattern.

The iterative approach can be also used to improve position weight matrices. For example a PWM for splice-site signals computed from vertebrate genomes can be iteratively refined to obtain a PWM specific for human.

In general it seems that the simple iterative methods are suitable for improvement of patterns obtained by other methods or from different data. However, this approach is not sufficient to discover patterns without any prior knowledge.

From Iteration to PTAS

The Consensus Pattern problem is another formulation of pattern discovery. The problem is defined as follows: find a pattern P and one occurrence of P in each sequence so that the total number of mismatches over all occurrences is minimized.

The Consensus Pattern problem is NP-hard. It is unlikely that any polynomial time algorithm for such problems exists.

Because there is no algorithm guaranteed to find the best solution of the Consensus Pattern problem in a reasonable time, we may wish to have a guarantee that the cost of the found pattern (i.e., the total number of mismatches) is at most α times the cost of the optimal pattern. The *approximation ratio* is termed α. For example, if $\alpha = 2$ we are guaranteed to find a pattern that has at most twice as many mismatches as the best possible pattern.

For some problems it is possible to construct an algorithm that works for any α supplied. However, the smaller the approximation ratio, the longer the algorithm runs. This type of algorithm is called the *polynomial approximation scheme*, or PTAS. The PTAS for a Consensus Pattern problem is based on a simple iterative idea repeated many times with different initial patterns.

The PTAS requires input sequences, the desired length L of a pattern, and a parameter r. It finds all possible combinations of r substrings of length L taken from input sequences. Each combination may contain zero, one, or several substrings from each sequence, some substrings may even repeat more than once. If the total length of all sequences is N, there are $O(N^r)$ combinations. For each combination of r substrings the following steps are performed:

- The majority pattern P of the r substrings is computed. This pattern has in each position the character occurring most frequently in this position in the r substrings.
- Find the best occurrence of P in each input sequence.
- Compute a new majority pattern P'.
- Find the best occurrences of P' in all sequences and compute the number of mismatches (cost of P').

The result will be the pattern P', which achieves the minimum cost. Notice, that the algorithm performs one step of iteration with the pattern obtained from each possible combination. The running time of the algorithm is $O(N^{r+1} L)$ and its approximation ratio is $1 + (4|\Sigma|A - 4) / \left[\sqrt{e}\left(\sqrt{4r+1} - 3\right)\right]$ for r 3.

This result is very interesting from the point of view of theoretical computer science. However the algorithm is not very practical. For example, if we choose $r = 3$ and $\Sigma = \{A,C,G,T\}$, the algorithm will identify the pattern with at most 13 times as many mismatches as the optimal pattern. The running time is $O(N^4L)$, impractical for large inputs. In order to achieve $\alpha = 2$, r needs to be at least 21. This gives an algorithm a prohibitive running time $O(N^{22})$. Of course, the approximation ratio is only an upper bound of the possible error. For some inputs the optimal or close-to-optimal results can be obtained even for small r, however there is no guarantee.

A program called COPIA is based on the ideas of PTAS. Several changes considerably reduce the running time. The enumeration of all possible combinations of r substrings is replaced by randomly sampling combinations. The consensus pattern obtained from each randomly chosen combination of substrings is improved by the iterative method until there is no further improvement. COPIA runs in reasonable time for real data but it does not have the same guarantees of pattern quality as the PTAS algorithm.

Machine Learning Methods

Sometimes a pattern cannot be described well by a simple deterministic pattern and one may wish to express it in a form of a stochastic model, such as Hidden Markov Model or position weight matrix (which is a simpler version of HMM). This kind of pattern is discovered using iterative expectation maximization techniques that do not necessarily converge to the global maximum.

Expectation Maximization

First we will consider a simpler case of position weight matrices. A simple learning algorithm called *expectation maximization* (EM) is used to estimate parameters of the stochastic model of a pattern that occurs once at an unknown position in each input sequence. The algorithm can be easily extended to more complicated models, e.g., patterns with flexible gaps, a finite mixture model.

The algorithm is iterative. It starts with some initial model parameters (usually randomly set). Each iteration consists of two steps as follows:

- **E step.** For every sequence s and for every position in s compute the probability that the occurrence of the pattern in s starts at this position. The probability is based on the model from the previous iteration (or initial model for the first iteration).

- **M step.** For every position in the pattern compute new probabilities for characters at this position. This is based on all possible occurrences of the pattern weighted by probabilities computed in E step. These values will form new parameters of the model.

Notice that the algorithm uses all possible occurrences of the pattern to obtain a new matrix, instead of only one occurrence in each string. Similarly to other iterative methods, the EM algorithm converges to a local maximum depending on the initial parameters of the model, instead of the global maximum likelihood. There is the additional assumption that every pattern occurs exactly once in every sequence.

These are addressed in the MEME algorithm, which is a modification of EM algorithm. The algorithm is based on the assumption that the pattern found should closely resemble at least one subsequence found in the dataset. Several or no possible occurrences of the pattern in a sequence could be considered. The algorithm proceeds as follows:

1. Form an initial model for each subsequence in the dataset. The initial model is a position weight matrix. For every position the character at the corresponding position in the subsequence has probability p (p is usually between 0.5 and 0.8), and all other characters have probability $(1 - p) / (|\Sigma| - 1)$.
2. One iteration of the EM algorithm is performed on each initial model. The likelihood score is computed for the resulting models.
3. The model with the largest likelihood score is selected as an initial model for the EM algorithm.

The algorithm can be forced to report more patterns by erasing all occurrences of the found pattern from the dataset and rerunning the entire process.

Hidden Markov Models

Hidden Markov Models (HMMs) can be used to model a family of sequences. Given an HMM and a sequence, it is possible to compute the most probable path through the model for this sequence in $O(nm)$ time using the Viterbi algorithm, where n is the length of the sequence and m is the number of states in the model. This path represents the most probable occurrence of the pattern in the sequence. The probability P that the sequence was generated by the model can also be computed in $O(nm)$ time using the forward algorithm. Value $-\log P$ is called a *NLL score of the sequence* with respect to the model. A higher probability of generating the sequence corresponds to a lower NLL score of the sequence.

There are three issues that need to be addressed if HMMs are used to represent a sequence family:

- **Topology of HMM.** Topology specifies the scheme of the Hidden Markov model that is used to represent a sequence family.
- **Training process.** The training process is needed to estimate the parameters of the model so that the sum of scores of sequences in the family is optimized.
- **Search for sequences.** The searching process should allow one to distinguish between sequences that belong to the family and sequences that do not.

Topology of HMM

A common HMM topology for sequence analysis is depicted in Fig. 4. The model consists of three types of states. *Match motif states* model conserved parts of the sequence. Match states specify the probability distribution of characters at each conserved position. There can be any number of match states in the model. We assume that the user gives this number beforehand. *Insertion states* model possible gaps between match states. Gaps can be arbitrarily long. The probability assigned to a self-loop in an insertion state determines probability distribution of possible gap lengths. The probability distribution is geometric, and the mean value can be easily computed. Finally, *deletion states* permit one to model occurrences of the pattern that do not contain some of the conserved positions.

Training of the Model

Given a topology of an HMM and a family of sequences to be modeled by the HMM, we can estimate the parameters of the model so that the model will generate sequences similar to those in the family with high probability.

The Baum-Welch algorithm can be used to perform this task. It is an iterative algorithm very similar to the EM algorithm used to estimate parameters of PWMs. We start with arbitrary parameters. If we have some prior knowledge of the sequence family, we can use this knowledge to set the initial parameters. Then in each step, the probabilities of all paths for all sequences are computed, and the model parameters are reestimated to minimize the NLL score of the training sequences to maximize the probability.

The algorithm does not guarantee a global optimum. It converges to a local minimum dependent on the initial parameter settings.

Search for Sequences

Searching for the pattern in the form of HMM is more complicated than in the case of simpler patterns. Given a sequence one can efficiently compute the most probable alignment of the sequence to the pattern and compute its NLL score. The NLL score gives a measure of how well the sequence can be aligned to the pattern. However, NLL scores depend highly on sequence length. In general, shorter sequences have smaller NLL scores than longer ones. Therefore, we cannot use a fixed threshold on an NLL score to discriminate between members and nonmembers of the family. This is possible only if all the input sequences have approximately the same length. Fortunately it was observed that sequences, that do not belong to the sequence family, form a line corresponding to a linear dependency. NLL scores for family members drop well below this line. Thus, it is possible to identify members of the family by a statistical test using a z-score for some window of sequence length. In order to estimate parameters needed to determine the z-score we need to compute the NLL score for many background sequences of different lengths not belonging to the family.

Enhancing HMM Models

Reducing the Number of Parameters

The greater the number of parameters of the model, the greater the amount of data required to properly train the model. Too many parameters can cause overfitting, where the model fits the training data very well but does not generalize to new sequences. There are several ways to reduce the number of parameters of the model. These include model surgery and different initial topology.

- **Model surgery.** Model surgery adjusts the model topology during the training in order to reduce the number of parameters of the model. In particular, this technique avoids two common problems that arise during training:

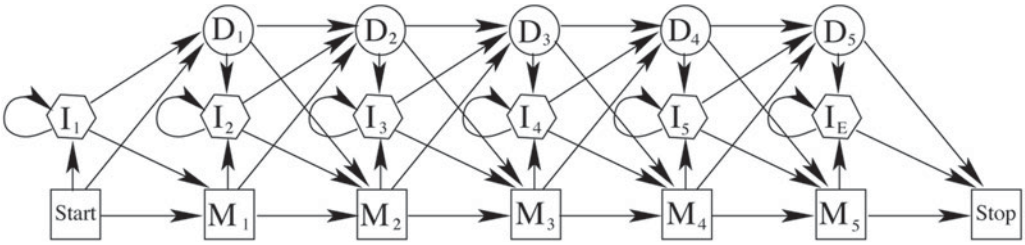


Fig. 4. A common HMM topology for pattern discovery. States $M_1, \ldots, M_5$ are match states, $I_1, \ldots, I_E$ are insertion states, and $D_1, \ldots, D_5$ are deletion states.

  - Some match states are used only by a few sequences. If the number of sequences using a match state drops below a given threshold (typically one-half), the state is removed. In this way we force sequences to either use the insertion state at this point, or significantly change their alignment.
  - Some insertion states are used by too many sequences. If an insertion state is used by more sequences than a given threshold (typically one-half), then the state is replaced by a chain of match states. The number of inserted match states is equal to the expected number of insertions in the replaced insertion state.
- **Different initial topology.** Meta-MEME uses a different program to report simple short patterns in the sequence. These patterns are transformed into matching states in HMM. The patterns are combined together using insertion states as shown in Fig. 5.

Discovering Subfamilies

Sometimes a family of sequences consists of several subfamilies. In such cases the family is represented by several motifs rather than by one. This problem can be solved by combining several HMMs with standard topology to one larger HMM as shown in Fig. 6.

If we do not have any preliminary knowledge of how to set the initial parameters of this model, it might be difficult to accurately train the model. In this case it is appropriate to use the Viterbi algorithm for training. The Viterbi training algorithm uses only the best path to reestimate the parameters, in contrast to the Baum-Welch algorithm that uses all paths weighted by their probabilities. Therefore, once the parameters of a part of the model reflect a bias to one subfamily, only sequences in this subfamily are used to train this part of the model in the Viterbi algorithm.

Methods Using Additional Information

Many biologically significant patterns are difficult to discover in sequences. In such cases additional sources of information can be used to guide the search.

Identifying Motifs in Aligned Sequences

Pattern discovery and multiple local alignments are closely related tasks. One can easily obtain a pattern (e.g., in a form of consensus sequence and PWM) from a given local alignment by taking each column of the alignment as one position of the pattern. The question begins to be interesting if we assume that the input contains errors (i.e.,

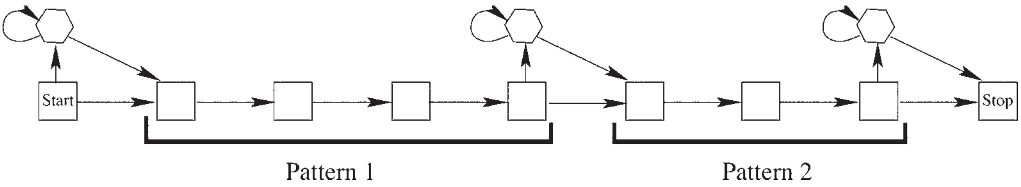


Fig. 5. Meta-MEME uses much simpler initial topology. Patterns found by other programs are connected together by insertion states.

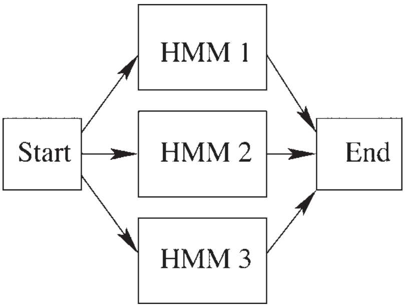


Fig. 6. Topology of HMM suitable for representing families of sequences with several subfamilies. HMM 1, 2, and 3 are models representing individual subfamilies.

some sequences are not aligned correctly) or it contains several subfamilies. In this case we may try to identify a pattern that does not match all sequences. This task is addressed by EMOTIF.

EMOTIF searches for motifs containing characters, ambiguous characters, and wildcards. The set of possible ambiguous characters is fixed. Each character (normal, ambiguous, or wildcard) corresponds to one column of the local alignment. It is possible to compute the specificity of each pattern of this form (how likely it is to occur by random) and sensitivity (how many training sequences it covers). EMOTIF identifies many motifs with different values of specificity and coverage. For several motifs with the same specificity, only the motif with the greatest coverage is reported and vice versa. Motifs are found by a pruned exhaustive search.

Global Properties of a Sequence

Computational recognition of eukaryotic promoters is a difficult task. In addition to local information in the form of transcription factor binding sites, it is also necessary to consider global properties of the DNA in the surrounding regions. One example are CpG islands: 1–2 kb long regions with higher frequency of CpG than found elsewhere in the genome. CpG islands often demarcate the 5' region flanking constitutively expressed vertebrate genes. In addition, regions downstream of genes usually have low flexibility (flexibility or bendability of DNA can be estimated from sequence-based models of DNA structure).

In the pattern matching problem we can use global information, such as CpG islands and flexibility, to distinguish random occurrences of a pattern from those that have functional significance. In pattern discovery we may use this kind of prior knowledge to choose appropriate parts of the genome as our input set to search for patterns.

Using Phylogenetic Tree

One of the basic assumptions in identifying patterns in biological sequences is that regions conserved in evolution are functionally important. Therefore it is natural to use known phylogenetic relations among sequences to guide the pattern search.

Assume we want to identify a regulatory element. Instead of using regulatory regions from many co-regulated genes of the same species, we will use regulatory regions of the same gene taken from many related species. We assume that the evolutionary tree of these species is known. Now we may try to identify the short pattern best conserved during evolution.

The best-conserved pattern can be identified using a parsimony measure as follows. We are given the length of pattern k. We want to associate a sequence t_w of length k with each node w. In internal nodes, t_w can be an arbitrary string. In each leaf, t_w is required to be a subsequence of the input sequence and corresponds to an occurrence of the pattern. We want to minimize the sum of $d\ (t_v,t_w)$ over all tree edges (v,w), where $d\ (t_v,t_w)$ is a distance between strings t_v and t_w (in this case the number of substitutions).

The algorithm works as follows. First the tree is rooted in an arbitrary internal node. For each node w and each possible string t of length k let $d^*_w(t)$ be the best possible parsimony score that can be achieved in the subtree rooted at w provided that $t_w = t$ (i.e., string t is stored in node w). Scores $d^*_w(t)$ can be computed in a leaves-to-root fashion.

The scores are easily found for leaf nodes: if t is a substring of the input string associated with the leaf w, then $d^*_w(t) = 0$, otherwise the score will be . Once we know all scores for both children w_1 and w_2 of a certain internal node w, we can compute the scores for w. If we assume that the node w_1 stores sequence t_{w_1}, w_2 stores t_{w_2} and w stores t_w, then the parsimony of the subtree rooted at w will be:

$$\left[d^*_{w_1}\left(t_{w_1}\right) + d\left(t_{w_1},t_w\right)\right] + \left[d^*_{w_2}\left(t_{w_2}\right) + d\left(t_{w_2},t_w\right)\right]$$

For each possible t_w we want to find t_{w_1} and t_{w_2} that minimize this sum and store the sum as $d^*_w(t_w)$. After we compute scores for all nodes, we can retrieve the overall minimum parsimony as the smallest score computed for the root. We can use the stored intermediate results to reconstruct the entire optimal solution in a root-to-leaves manner. Strings of t_w stored in the leaves represent the occurrences of the pattern.

With additional optimizations the algorithm can be implemented to run in time $O(nk|\Sigma|^k)$, where n is the number of leaves. This is exponential in k but the problem is NP-hard, so we cannot hope to find a polynomial solution.

The use of phylogenetic tree can help to find patterns which are hard to find otherwise. For example, if the input contains several families of sequences, other methods may discover patterns characteristic for the sequences in the largest family instead of the patterns shared by all sequences. In practice, this problem is often solved by a preprocessing step: sequences are first clustered to separate families and then a single member is chosen from each family for further processing. If we use the phylogenetic tree, this step is not necessary and we keep information from all sequences. Because close homologs are grouped together in the phylogenetic tree, their weight is not so great, as the pattern still has to agree with other parts of the tree.

Use of Secondary/Tertiary Structure

Positions important to the secondary and tertiary structure of proteins are usually well conserved. If we know the structure of proteins in question, we can try to locate regions important for achieving this structure. These regions are good candidates to identify occurrences of our pattern. One possibility is to choose points of contact between two secondary structure elements as candidate spots for conserved positions. This approach has been used to construct a sparse deterministic pattern containing ambiguous characters and flexible gaps. If possible, sequences are aligned so that the points of contact align together. In such an alignment we can choose positions that are well-preserved among those columns that contain many points of contact.

Secondary structure and motif searching are also closely knit together in algorithms that identify conserved patterns in RNA sequences. RNA molecules are more related in their structure than in their sequence. Identifying RNA secondary structure of a set of related RNA sequences is best accomplished by first aligning the sequences. Alignment is then used to discover similarities. The process can be iterated resulting in simultaneous discovery of the alignment, secondary structure features, and conserved patterns.

Finding Homologies Between Two Sequences

Finding homologies between two DNA or protein sequences is a special case of the general pattern discovery problem. Here, the problem becomes simpler in principle but with larger amounts of data, the challenge shifts to efficiency and scalability. It is not our intention to survey the entire field of similarity searching, we will only consider the specific problem of comparing two very long genome sized DNA sequences, to shed some light on this problem.

In theory, this problem is easily solved by standard Smith-Waterman dynamic programming algorithm. However, when sequences are long, the dynamic programming and FASTA strategies become too expensive. Scalable heuristics are required.

Two strategies have lead to improvements. The first is exemplified by the popular Blast family of algorithms. This approach finds short exact *seed* matches (hits), which are then extended into longer alignments. However, when comparing two very long sequences, most programs in this category such as SIM, Blastn (BL2SEQ), WU-Blast, and Psi-Blast run slowly and require large amounts of memory. SENSEI is somewhat faster and uses less memory than the above, but it is currently limited to ungapped alignments. MegaBlast runs quite efficiently without a gap open penalty and a large seed length of 28 yielding much lower sensitivity.

Another strategy, exemplified by MUMmer, QUASAR, and REPuter, uses suffix trees. Suffix trees suffer from two problems: They were designed to deal with precise matches and are limited to comparing highly similar sequences. MUMmer and QUASAR have implemented various ways of linking precisely matched neighboring blocks. The second problem with suffix trees is that they have an intrinsic large space requirement.

A new program PatternHunter has recently been developed. PatternHunter uses optimized spaced seeds for high sensitivity and improved search and alignment algorithms. It is implemented in Java and runs at the speed of MegaBlast and the suffix tree program family while producing output at default Blastn sensitivity. PatternHunter was tested against the newly improved Blastn (using BL2SEQ) and MegaBlast, downloaded from the NCBI website on July 9, 2001. Program compari-

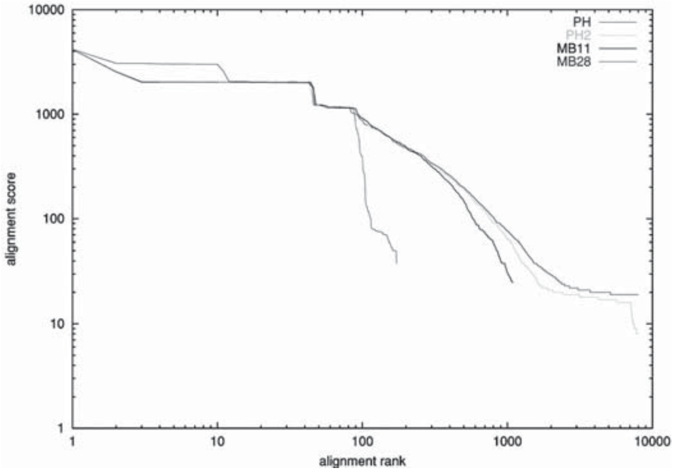


Fig. 7. Input: *H. influenza* and *E. coli*. The score is plotted as a function of the rank of the alignment, with both axes logarithmic. MegaBlast (MB28) misses over 700 alignments of score at least 100. MB11 is MegaBlast with seed size 11 (it is much slower and uses more memory), indicating the missed alignments by MB28 are mainly due to seed size.

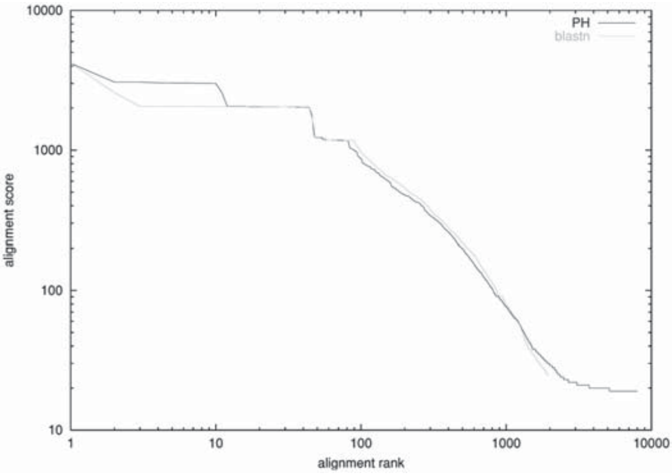


Fig. 8. Input: *H. influenza* and *E. coli*. PatternHunter produces better quality output than Blastn while running 20 times faster.

sons were performed on a 700 MHz Pentium III PC with 1 gbyte of memory. Experiments were performed on the following datasets: *M. pneumoniae* vs *M. genitalium*, *E. coli* vs *H. influenza*, and *A. thaliana* chromosome 2 vs *A. thaliana* chromosome 4 were then undertaken. All programs were run without filtering (bl2seq option –F F) to ensure identical input to the actual matching engines. With filter, Figs. 7 and 8 essentially remained the same. All comparisons were based on same scoring reward and penalties to insure the output results were comparable. MB28 is MegaBlast with seed size 28. PH is PatternHunter at sensitivity similar to Blastn seed size 10 , and PH2 is PatternHunter at sensitivity similar to Blastn seed size 11. Figures 7–9 show the output quality of PatternHunter vs Blastn and MegaBlast. In Fig. 7, MegaBlast using seed

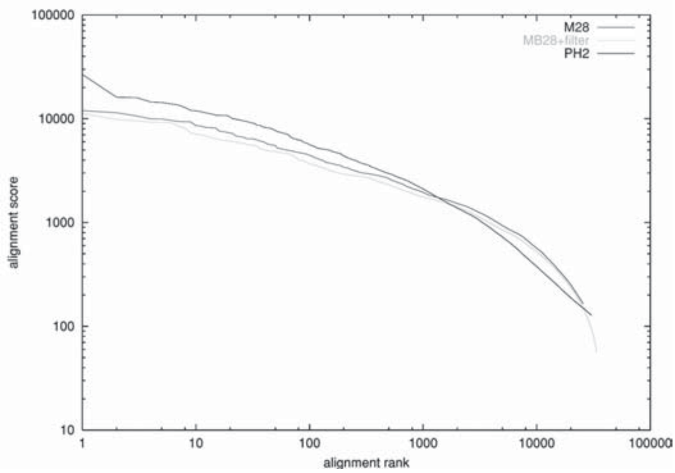


Fig. 9. Input: *A. thaliana* chr 2 and chr 4. PatternHunter (PH2) outscores MegaBlast in one sixth of the time and one quarter the memory. Both programs used MegaBlast's non-affine gap costs (with gapopen 0, gapextend –7, match 2, and mismatch –6) to avoid MegaBlast from running out of memory. For comparison we also show the curve for MegaBlast with its default low complexity filtering on, which decreases its runtime more than sixfold to 3305 seconds.

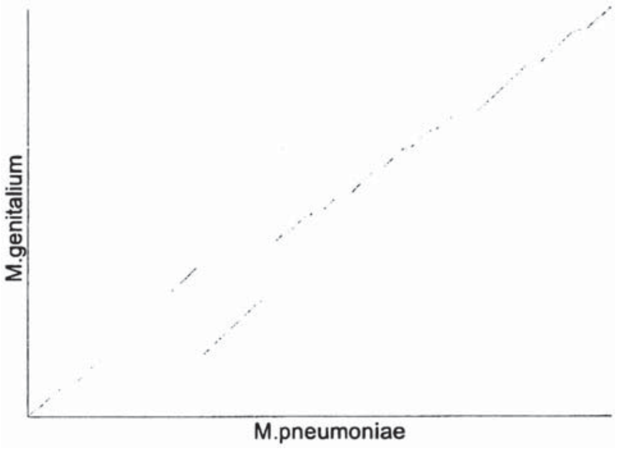


Fig. 10. Highscoring Segment Pairs plot of *M. Pneumoniae* vs *M. Genitalium*.

weight 28 (MB28) misses over 700 high scoring alignments. Using the same parameters, PatternHunter results are comparable or better than results obtained by Blastn. It is 20 times faster and uses one tenth the memory (*see* Fig. 8). Figure 9 shows that MegaBlast produces alignments with significantly lower scores compared to PatternHunter (PH2), which uses only one sixth the time and one quarter the space, on arabidopsis chromosomes. Figure 10 shows a genome alignment of *M. Pneumoniae* vs *M. Genitalium*, by PatternHunter and MUMmer. The table in Figure 11 compares the time and space used by PatternHunter (PH2). For example, the comparison of human chromosome 22 (35M bases) vs human chromosome 21 (26.2M bases) only required 1 h to complete.

Seq1	*Size*	*Seq2*	*Size*	*PH*	*PH2*	*MB28*	*Blastn*
M. pneumoniae	828K	*M. genitalium*	589K	10s/65M	4s/48M	1s/88M	47s/45M
E. coli	4.7M	*H. influenza*	1.8M	34s/78M	14s/68M	5s/561M	716s/158M
A. thaliana chr 2	19.6M	*A. thaliana* chr 4	17.5M	5020s/279M	498s/231M	21720s/1087M	

Fig. 11. Performance comparison: If not specified, all with match 1, mismatch –1, gap open –5, gap extension –1. Table entries under PH, PH2, MB28 and Blastn indicate time (seconds) and space (megabytes) used;   means out of memory or segmentation fault.

Assessment of Pattern Quality

A wide variety of pattern discovery methods are available. They differ in what they consider to be the best pattern and most are not guaranteed to identify the best pattern. Thus we need to evaluate the quality of the discovered patterns. Various statistical methods have been employed to address the question: how likely is it that the pattern occurs in our sequences merely by chance? The smaller the likelihood the bigger the chance that the pattern discovered has biological meaning. Statistical significance of a pattern is used to evaluate the performance of different algorithms, but also as a scoring function to rank discovered patterns for a user, or to guide a search for the most significant pattern.

Although statistical significance is an important tool that allows one to distinguish artifacts of search algorithms from significant patterns, one must keep in mind that the goal of pattern discovery is to identify elements of certain biological importance. Even a very significant pattern may not be what we are looking for. For example we may want to discover functionally important sites but the pattern was conserved because it was essential for structure instead. Therefore it is important to verify patterns by appropriate biological experiments. For example, mutagenesis or another appropriate biological assay can be used to verify the function of a protein, x-ray crystallography or nuclear magnetic resonance (NMR) to determine the structure of a protein, and DNA footprinting to verify binding sites.

Background Model

If we want to understand what the probability is that a pattern occurs by random we need to define *random*. In other words, we need to select a background model. The simplest background model assumes that all possible characters of the alphabet are equally likely and the individual positions of the sequence are independent. Therefore all possible sequences of characters of the same length are equally likely.

This model is usually not adequate because different characters of the alphabet, i.e., individual nucleotides or amino acids, occur in biological sequences with different frequencies. For example in an AT rich sequence we can expect that the string 'TAATA' will be more frequent that 'CGGCG'. We can solve this problem using the *Bernoulli model*. In this model each character of the alphabet potentially has a different probability but individual positions of the sequence are still independent.

An even more complicated model is *Markov chain*. In this case, the probability of each character at position j depends on the characters at positions $j – 1, j – 2,..., j – k$, where k is a parameter called the order of the Markov chain. A Markov chain of order 0 is identical to the Bernoulli model. Markov chains take into account the fact that some combinations of characters occur less or more frequently than expected based on the frequencies of their constituents.

Parameters of background models, i.e., probabilities of individual characters, can be estimated as a function of the observed frequencies in the input sequences or in some larger databases.

Pattern Significance

Given a deterministic pattern P and a sequence of length L we can simply count the number of occurrences of P in the sequence. Denote this number N_P. Let $E(X_{P,L})$ be the expected number of occurrences of pattern P in a sequence of length L generated by a background model. If the observed number of occurrences N_P is much higher than the expected value $E(X_{P,L})$, P is then a significant pattern. The standard measure used in this context is the z-score:

$$z_P = \frac{N_P - E\left(X_{P,L}\right)}{\sigma\left(X_{P,L}\right)}$$

where $\sigma(X_{P,L})$ is the standard deviation of the number of occurrences of P in a random sequence of length L. This measure gives the number of standard deviations by which the observed value N_P differs from the expected value.

The simplest approach towards computing the z-score is to generate a large number of sequences using the chosen background model, count the number of occurrences of the pattern, then estimate the expected value and standard deviation from this random sample. This works for any kind of background model and pattern but it has a high running time (especially if we need to evaluate the z-score for many patterns). The values obtained are only estimates.

Another approach is to compute the mean and variance of the distribution of the number of occurrences of a given pattern exactly. For example, algorithms are available for a large class of patterns when the background distribution is a Markov chain of order k.

Information Content

In case of probabilistic models such as position weight matrices we do not have strictly defined occurrences but rather a score between 0 and 1 for any string. One can set a threshold on what we consider to be an occurrence and then evaluate the z-score or other appropriate statistical measures. Alternatively when evaluating position weight matrices an *information content* (also called *relative entropy*) measure is used. This tells how much the distribution defined by the PWM differs from the (Bernoulli-type) background distribution. Relative entropy is computed as follows:

$$\sum_i \sum_c A[c,i] \log_2 \frac{A[c,i]}{f(c)}$$

where $A[c,i]$ is the frequency of character c in column i of the matrix and $f(c)$ is the background frequency of the character c. Relative entropy has two disadvantages. First, it does not depend on the number of occurrences of the pattern in the sequences. A strong pattern with very many occurrences has higher relative entropy than a weaker pattern with few occurrences. Therefore, it is an appropriate measure only in situations where the pattern is required to occur in all sequences. This is often the case in Gibbs sampling methods. Second, relative entropy of one column is always non-negative and therefore if we add columns that are not well-conserved to the pattern, we can obtain a better score. Therefore relative entropy is not suitable for comparing patterns of different lengths. This can be solved by subtracting the appropriate term from the contribution of each column so that the expected contribution of a column is zero.

Sensitivity and Specificity of Classification

One application of the various methods of pattern discovery is to identify patterns that characterize a given family of related sequences. In this context we need to measure how well we can distinguish members of the family from nonmembers based on the occurrence of the pattern. For this purpose a test set consisting of sequences of a known family is required. We find all occurrences of the motif in the test set and compute the following four scores: TP (true positives) are sequences that contain the motif and belong to the family in question, TN (true negatives) are sequences that do not belong to the family and do not contain the motif, FP (false positives) are sequences that contain the motif but do not belong to the family and FN (false negatives) are sequences that do not contain the motif but belong to the family. Thus $TP + TN$ is the number of correct predictions and $FN + FP$ is the number of wrong predictions. Based on counts of TP, TN, FP, FN we can define various measures. Sensitivity (also called *coverage*) is defined as $TP / (TP + FN)$ and specificity is defined as $TN / (TN + FP)$. A pattern has maximum sensitivity, if it occurs in all proteins in the family (regardless of the number of false positives) and it has maximum specificity, if it does not occur in any sequence outside the family. A score called *correlation coefficient* gives an overall measure of prediction success:

$$C = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)\,(TP + FN)\,(TN + FN)\,(TN + FP)}}$$

This expression grows from –1 to 1 as the number of correct predictions increases.

Concluding Remarks

An overview of methods available for pattern discovery was presented. The tools developed by computer scientists are common today in many biological laboratories. They are required to handle large-scale tasks, including the annotation of newly sequenced genomes, organization of proteins into families of related sequences, or identifying regulatory elements in co-expressed genes. They are also important in smaller-scale projects because they can be used to detect possible sites of interest and assign putative structure or function. Thus, they can be used to guide biological experiments in *wet labs*, decreasing the time and money spent in discovering new biological knowledge.

Indeed, there are many examples, where computational tools have helped biologists to make important discoveries. For example, pattern-discovery tools helped to identify a number of putative secretory proteins in *Mycobacterium tuberculosis* genome. Subsequently, 90% of the predicted candidates were experimentally confirmed. Identification of *M. tuberculosis* secretory proteins is a first step to the design of more effective vaccines against tuberculosis.

In order to fully understand the meaning of the output of a pattern-discovery tool, biologists need to understand the basics of the algorithm used in the tool. It is very useful to know the performance guarantees of the algorithm. Tools that cannot guarantee finding the best or all patterns, might find only low-scoring patterns in sequences which contain high-scoring patterns as well.

The pattern discovery process is often a computationally intensive task. Therefore many databases are maintained and updated containing results of pattern discovery applied to particular tasks. These databases often contain experimental evidence from biological literature and other useful information. On the accompanying CD-ROM, we provide a list of such databases together with related links and short descriptions of database contents. An overview of the software tools is also included on the CD-ROM.

Acknowledgments

Parts of this chapter are based on technical report of Brejová et al. (2000) and some material from paper of Ma et al. (2002). We would like to thank co-authors of the report Chrysanne DiMarco, Sandra Romero Hidalgo, Gina Holguin, and Cheryl Patten and co-authors of the paper Bin Ma and John Tromp for cooperation and helpful discussion. We want to also thank Stephen Krawetz, Jonathan Badger, Paul Kearney, and Marthenn Salazar who kindly reviewed parts of the material.

Suggested Readings

Biological Motivation and Introductory Reading

Batzoglou, S., Pachter, L., Mesirov, J. P., Berger, B., and Lander, E. S. (2000) Human and mouse gene structure: comparative analysis and application to exon prediction, Genome Res. 10(7), 950–958.

Fickett, J. W. and Hatzigeorgiou, A. G. (1997) Eukaryotic promoter recognition, Genome Res. 7(9), 861–868.

Gelfand, M. S., Koonin, E. V., and Mironov, A. A. (2000) Prediction of transcription regulatory sites in Archaea by a comparative genomic approach, Nucleic Acids Res. 28(3), 695–705.

Gomez, M., Johnson, S., and Gennaro, M. L. (2000) Identification of secreted proteins of Mycobacterium tuberculosis by a bioinformatic approach, Infect. Immun. 68(4), 2323–2327.

Hardison, R. C., Oeltjen, J., and Miller, W. (1997) Long human-mouse sequence alignments reveal novel regulatory elements: a reason to sequence the mouse genome, Genome Res. 7(10), 959–966.

Hughes, J. D., Estep, P. W., Tavazoie, S., and Church, G. M. (2000) Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae, J. Mol. Biol. 296(5), 1205–1214.

Linial, M., Linial, N., Tishby, N., and Yona, G. (1997) Global self-organization of all known protein sequences reveals inherent biological signatures, J. Mol. Biol. 268(2), 539–546.

Mironov, A. A., Koonin, E. V., Roytberg, M. A., and Gelfand, M. S. (1999) Computer analysis of transcription regulatory patterns in completely sequenced bacterial genomes, Nucleic Acids Res. 27(14), 2981–2989.

Riechmann, J. L., Heard, J., Martin, G., Reuber, L., Jiang, C., Keddie, J., et al. (2000) Arabidopsis transcription factors: genome-wide comparative analysis among eukaryotes, Science 290(5499), 2105–2110.

Yada, T., Totoki, Y., Ishii, T., and Nakai, K. (1997) Functional prediction of B. subtilis genes from their regulatory sequences, in: Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology (ISMB) (Gaasterland, T., Karp, P., Ouzounis, C., Sander, C., and Valencia, A., eds.) The AAAI Press, Halkidiki, Greece, pp. 354–357.

Related Books and Surveys

Brazma, A., Jonassen, I., Eidhammer, I., and Gilbert, D. (1998) Approaches to the automatic discovery of patterns in biosequences, J. Comp. Biol. 5(2), 279–305.

Brejová, B., DiMarco, C., Vinar, T., Hidalgo, S. R., Holguin, G., and Patten, C. (2000) Finding Patterns in Biological Sequences, Technical Report CS-2000-22, Dept. of Computer Science, University of Waterloo, Ontario, Canada.

Gusfield, D. (1997) Algorithms on strings, trees and sequences: computer science and computational biology, Chapman & Hall, New York, NY.

Pevzner, P. A. (2000) Computational molecular biology: an algorithmic approach, The MIT Press, Cambridge, MA.

Rigoutsos, I., Floratos, A., Parida, L., Gao, Y., and Platt, D. (2000) The emergence of pattern discovery techniques in computational biology, Metabolic Eng. 2(3), 159–167.

Pattern Discovery

Gorodkin, J., Heyer, L. J., Brunak, S., and Stormo, G. D. (1997) Displaying the information contents of structural RNA alignments: the structure logos, Comp. Appl. Biosci. 13(6), 583–586.

Schneider, T. D. and Stephens, R. M. (1990) Sequence logos: a new way to display consensus sequences, Nucleic Acids Res. 18(20), 6097–6100.

Algorithms for Pattern Discovery

Exhaustive Search Methods

Jonassen, I. (1996) Efficient discovery of conserved patterns using a pattern graph, Technical Report 118, Department of Informatics, University of Bergen, Norway.

Parida, L., Rigoutsos, I., Floratos, A., Platt, D., and Gao, Y. (2000) Pattern discovery on character sets and real-valued data: linear bound on irredundant motifs and an efficient polynomial time algorithm, in: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM Press, San Francisco, CA, pp. 297–308.

Pevzner, P. A. and Sze, S. H. (2000) Combinatorial approaches to finding subtle signals in DNA sequences, in: Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB), (Bourne, P., Gribskov, M., Altman, R., Jensen, N., Hope, D., Lengauer, T., et al., eds.) The AAAI Press, San Diego, CA, pp. 269–278.

Rigoutsos, I. and Floratos, A. (1998) Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm, Bioinformatics 14(1), 55–67. Published erratum appears in Bioinformatics, 14(2), 229.

Rigoutsos, I. and Floratos, A. (1998) Motif discovery without alignment or enumeration (extended abstract), in: Proceedings of the 2nd Annual International Conference on Computational Molecular Biology (RECOMB), (Istrail, S., Pevzner, P., Waterman, M., eds.) ACM Press, New York, NY, pp. 221–227.

Smith, H. O., Annau, T. M., and Chandrasegaran, S. (1990) Finding sequence motifs in groups of functionally related proteins, Proc. Natl. Acad. Sci. USA 87(2), 826–830.

Tompa, M. (1999) An exact method for finding short motifs in sequences, with application to the ribosome binding site problem, in: Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB), (Glasgow, J., Littlejohn, T., Major, F., Lathrop, R., Sankoff, D., and Sensen, C., eds.) The AAAI Press, Montreal, Canada, pp. 262–271.

van Helden, J., Andre, B., and Collado-Vides, J. (1998) Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies, J. Mol. Biol. 281(5), 827–832.

Iterative Methods

Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., and Wootton, J. C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment, Science 262(5131), 208–214.

Li, M., Ma, B., and Wang, L. (1999) Finding Similar Regions in Many Strings, in: Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC), Atlanta, ACM Press, Portland, OR, pp. 473–482.

Liang, C. (2001) COPIA: A New Software for Finding Consensus Patterns in Unaligned Protein Sequences. Master thesis, University of Waterloo.

Liu, J. S., Neuwald, A. F., and Lawrence, C. E. (1995) Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies, J. Am. Stat. Assoc. 90(432), 1156–1170.

Neuwald, A. F., Liu, J. S., Lipman, D. J., and Lawrence, C. E. (1997) Extracting protein alignment models from the sequence database, Nucleic Acids Res. 25(9), 1665–1667.

Singh, M., Berger, B., Kim, P. S., Berger, J. M., and Cochran, A. G. (1998) Computational learning reveals coiled coil-like motifs in histidine kinase linker domains, Proc. Natl. Acad. Sci. USA 95(6), 2738–2743.

Zhang, M. Q. (1998) Statistical features of human exons and their flanking regions, Human Mol. Genet. 7(5), 919–922.

Machine Learning Methods

Bailey, T. L. and Elkan, C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers, in: Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB), (Altman, R., Brutlag, D., Karp, P., Lathrop, R., and Searls, D., eds.) The AAAI Press, Stanford, CA, pp. 28–36.

Bailey, T. L. and Elkan, C. (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization, Machine Learning 21(1/2), 51–80.

Lawrence, C. E. and Reilly, A. A. (1990) An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences, Proteins, 7(1), 41–51.

Hidden Markov Models

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. (1998), Biological Sequence Analysis, Cambridge University Press, Cambridge, UK.

Grundy, W. N., Bailey, T. L., Elkan, C. P., and Baker, M. E. (1997) Meta-MEME: motif-based hidden Markov models of protein families, Comp. Appl. Biosci. 13(4), 397–406.

Hughey, R. and Krogh, A. (1996) Hidden Markov models for sequence analysis: extension and analysis of the basic method, Comp. Appl. Biosci. 12(2), 95–107.

Krogh, A., Brown, M., Mian, I. S., Sjolander, K., and Haussler, D. (1994) Hidden Markov models in computational biology. Applications to protein modeling, J. Mol. Biol. 235(5), 1501–1501.

Pattern Discovery Using Additional Information

Blanchette, M., Schwikowski, B., and Tompa, M. (2000) An exact algorithm to identify motifs in orthologous sequences from multiple species, in: Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB), (Bourne, P., Gribskov, M., Altman, R., Jensen, N., Hope, D., Lengauer, T., et al., eds.) The AAAI Press, San Diego, CA, pp. 37–45.

Chiang, D. Y., Brown, P. O., and Eisen, M. B. (2001), Visualizing associations between genome sequences and gene expression data using genome-mean expression profiles, Bioinformatics 17(S1), S49–S55.

Eidhammer, I., Jonassen, I., and Taylor, W. R. (2000) Structure comparison and structure patterns, J. Comp. Biol. 7(5), 685–716.

Gorodkin, J., Heyer, L. J., and Stormo, G. D. (1997b) Finding the most significant common sequence and structure motifs in a set of RNA sequences, Nucleic Acids Res. 25(18), 3724–3732.

Ison, J. C., Blades, M. J., Bleasby, A. J., Daniel, S. C., Parish, J. H., and Findlay, J. B. (2000) Key residues approach to the definition of protein families and analysis of sparse family signatures, Proteins 40(2), 330–331.

Nevill-Manning, C. G., Wu, T. D., and Brutlag, D. L. (1998) Highly specific protein sequence motifs for genome analysis, Proc. Natl. Acad. Sci. USA 95(11), 5865–5871.

Pedersen, A. G., Baldi, P., Chauvin, Y., and Brunak, S. (1999) The biology of eukaryotic promoter prediction-a review, Comp. Chem. 23(3–4), 191–207.

Finding Homologies Between Two Sequences

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25(17), 3389–3392.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic local alignment search tool, J. Mol. Biol. 215(3), 403–410.

Burkhardt, S., Crauser, A., Ferragina, P., Lenhof, H.-P., Rivals, E., and Vingron, M. (1999) q-gram based database searching using a suffix array (QUASAR), in: Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB), ACM Press, Lyon, France, pp. 77–83.

Delcher, A. L., Kasif, S., Fleischmann, R. D., Peterson, J., White, O., and Salzberg, S. L. (1999) Alignment of whole genomes, Nucleic Acids Res. 27(11), 2369–2376.

WWW Gish, W. (2001) WU-Blast website (*see* Website: http://blast.wustl.edu/).

Huang, X. and Miller, W. (1991) A time-efficient, linear-space local similarity algorithm, Adv. Appl. Math. 12(3), 337–357.
WWW (*see* SIM Website: http://www.expasy.ch/tools/sim.html)

Kurtz, S. and Schleiermacher, C. (1999) REPuter: fast computation of maximal repeats in complete genomes, Bioinformatics 15(5), 426–427.

Lipman, D. J. and Pearson, W. R. (1985) Rapid and sensitive protein similarity searches, Science 227(4693), 1435–1441.

Ma, B., Tromp, J., and Li, M. (2002) PatternHunter faster and more sensitive homology search, Bioinformatics 18(3), 440–445.

Smith, T. F. and Waterman, M. S. (1981) Identification of common molecular subsequences, J. Mol. Biol. 147(1), 195–197.

States, D. J. and Agarwal, P. (1996) Compact encoding strategies for DNA sequence similarity search, in: Proceedings of the 4th International Conference on Intelligent Systems for Molecular Biology (ISMB), (States, D. J., Agarwal, P., Gaasterland, T., Hunter, L., and Smith, R. F., eds.) The AAAI Press, St. Louis, MO, pp. 211–217.

(*see* SENSEI Website: http:<stateslab.wustl.edu/software/sensei/).

Tatusova, T. A. and Madden, T. L. (1999) BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences, FEMS Microbiol. Lett. 174(2), 247–250.

Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. (2000) A greedy algorithm for aligning DNA sequences, J. Comp. Biol. 7(1–2), 203–204.

Assessment of Pattern Quality

Nicodème, P., Salvy, B., and Flajolet, P. (1999) Motif statistics, in: Algorithms — ESA '99, 7th Annual European Symposium, vol. 1643, Lecture Notes in Computer Science, (Nesetril, J., ed.), Springer, Prague, pp. 194–211.

Pesole, G., Liuni, S., and D'Souza, M. (2000) PatSearch: a pattern matcher software that finds functional elements in nucleotide and protein sequences and assesses their statistical significance, Bioinformatics 16(5), 439–440.

Rocke, E. and Tompa, M. (1998) An algorithm for finding novel gapped motifs in DNA sequences, in: Proceedings of the 2nd Annual International Conference on Computational Molecular Biology (RECOMB), (Istrail, S., Pevzner, P., and Waterman, M., eds.), ACM Press, New York, NY, pp. 228–233.

30 The Role of Transcription Factor Binding Sites in Promoters and Their *In Silico* Detection

Thomas Werner

Introduction

Transcription is the general process of copying part of the genomic DNA into RNA. This RNA either has functions of its own or is used to produce proteins. This is a multi-step process requiring the coordinated interaction of a plethora of proteins with the genomic DNA at various loci. The major regions of DNA involved in transcription are the regulatory regions like locus control regions, enhancers, and promoters.

Regulatory regions have to fulfill several requirements in order to serve a biological function in transcriptional control. The general chromatin status is a major determinant of sequence accessibility and is dependent on special protein complexes known as nucleosomes and their higher-order organization. On the molecular level, DNA methylation is of crucial importance and all processes discussed below can only occur after chromatin structure as well as DNA methylation have achieved a state where regulatory sequences are accessible for protein binding.

The basic principles of the interaction of proteins with DNA in transcriptional control are identical for all of these regions, so it is possible to focus on one representative region. The promoter is the best-studied of all regulatory regions and will be reviewed.

The Promoter

In general, the promoter is an integral part of the gene and often makes sense only in the context of its own gene, especially if important parts of the regulation are determined outside of the promoter (e.g., by an intron enhancer). The function of a promoter is to mediate and control initiation of transcription of that part of a gene that is located immediately downstream of the promoter (3'). This can be achieved either in an unregulated permanent manner (constitutive transcription) or in a highly regulated fashion by which transcription is subjected to the control of various extracellular and intracellular signals (regulated transcription). The DNA region required to fulfill this function can be determined by assays for promoter function in a heterologous context. I will refer to a promoter mainly as the region that is necessary to achieve transcriptional initiation, although this region may not be sufficient to completely determine the regulation of a gene. The promoter by definition marks the beginning of the first exon of a gene (*see* Fig. 1). The functional setup of promoters is intimately coupled

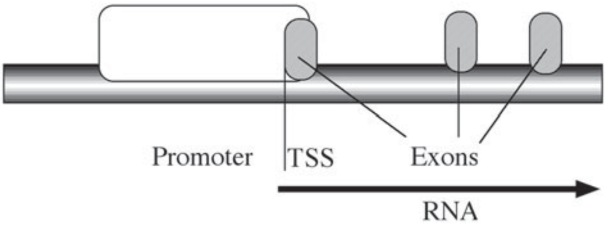


Fig. 1. Location of the promoter in genomic DNA. The promoter overlaps with the first exon and contains the transcription start site (TSS) where the transcription of genomic DNA into the nuclear RNA starts.

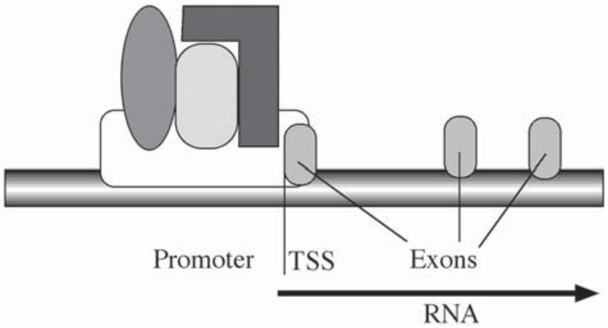


Fig. 2. Binding of transcriptional activators to the upstream promoter. The transcription factors bind to the promoter sequence and form the *"activation complex,"* which attracts the polymerase complex to the promoter.

with the basic events of transcriptional initiation. The focus throughout this chapter will be on polymerase II responsive promoters, which are controlling the vast majority of cellular genes encoding proteins in higher eukaryotes.

Transcription can only proceed after a competent transcription complex consisting of RNA polymerase II (pol II) and several general transcription factors (GTFs) have been recruited to the promoter. Usually, pol II is not capable of functionally interacting with a promoter alone and initiating transcription by itself and requires a host of cofactors. The polymerase complex is mainly concerned with accurately copying the DNA into RNA and not with determining where to start this process. This is the job of other types of proteins, called *activators*, that help to recruit the RNA polymerase to the correct location on the genomic DNA. Basically, there are two different phases in transcriptional activation of a gene. The first step includes a variety of transcription factors (TFs), that bind to upstream promoter and enhancer sequences to form a multiprotein complex (*see* Fig. 2). In the second step this complex directly or indirectly recruits a pol II complexed with GTFs to the core promoter and the transcription start site (TSS) located within the core promoter. Subsequently, transcription is initiated by this initiation complex, which itself is subject to regulatory influences of TFs.

Once the complete complex including the polymerase complex is assembled on the promoter (called the *initiation complex*), it is now competent to initiate RNA

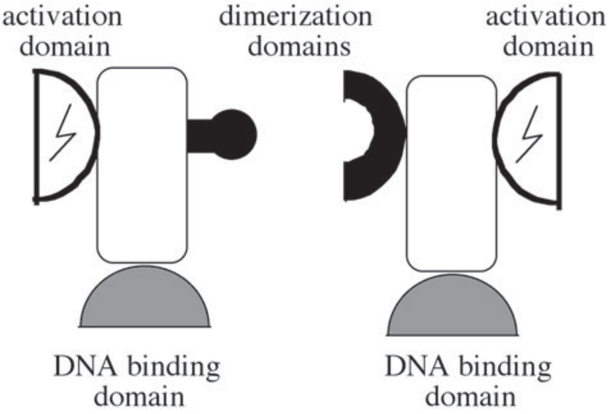


Fig. 3. Schematic representation of two transcription factors that can dimerize. The DNA binding domains may have different specificities as well as the activation domains that do not need to be of the same kind. The dimerization domains determine which factors can form dimers.

synthesis. The promoter can be defined as the minimum region of DNA that allows the formation of a functional initiation complex. This can be reduced to a core promoter or may include one or several upstream and/or downstream elements. Other regions directly upstream to this minimum promoter are termed *upstream regulatory regions* as long as they cannot be clearly identified as separate functional units (e.g., enhancers). There is a gradual transition from upstream promoter elements to enhancer elements (position and orientation independent activator regions), which is not necessarily obvious from inspecting the sequence. Similarly, the elements located downstream of the minimal promoter may also be downstream of the promoter elements and may constitute part of enhancers.

Transcription Factors

As evident, the key players in promoter activation are specialized proteins called *transcription factors*. Although there is a large variety of transcription factors in a single cell, at least several hundred if not a thousand, share some very basic features.

All transcription factors (TFs) contain a DNA binding domain enabling them to bind to genomic DNA as well as an activator domain, which has either an activating or suppressing activity. Many transcription factors have the ability to form specific homo- or hetero-dimers with other TFs. Such factors contain an additional domain, called a *dimerization domain* completing the generic structure of a TF shown in Fig. 3.

The function of a transcription factor is very simple and by itself not directly coupled to transcriptional control. The factor brings its action domain into a specific location by binding to the genomic DNA at selected sites. Whether this has consequences for transcription of a gene is determined by the functional context, i.e., by what other factors are bound or brought close enough to interact with the particular factor. Only a larger complex of transcription factors that often contain regulatory proteins that do not bind directly to DNA (so called *mediators of transcription*) have the ability to recruit the DNA polymerase complex to the promoter and thus influ-

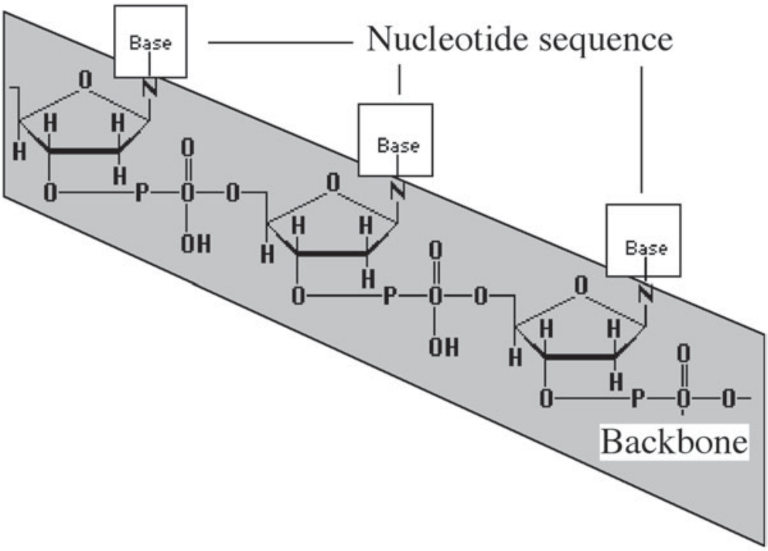


Fig. 4. Basic structure of a single DNA strand. The gray area represents the sugar-phosphate backbone supporting general protein contacts. The white boxes represent the bases that form sequence-specific contacts with proteins.

ence transcription. Transcription is a very dynamic process. Aggregate binding of a variety of factors to genomic DNA form an active initiation complex and the timely dissociation of the initiation complex, and even release of transcription factors from the DNA is crucial to transcription. This push/pull complicates considerably the process of optimization of binding site detection. It has very important consequences on the nature and detection of TF binding sites as will become evident below.

How Transcription Factors Bind to DNA

Transcription factors bind to DNA via a multitude of atomic interactions that are either van der Waals hydrophobic contacts or supported by juxtaposition of oppositely charged amino acids and DNA components. The two basic modes of molecular interactions can be distinguished:

1. The first mode involves nonspecific contacts between the protein side chains and the so-called *backbone* of the DNA, consisting of the sugar-phosphate structure that links the bases together. Such contacts can form anywhere on a DNA (double) strand and are responsible for the general tendency of TFs to associate with DNA.
2. The second mode is sequence-specific recognition. This is achieved by the direct contact of the amino acid side chains with the DNA bases. Therefore, these contacts can only be formed where there is a suitable succession of bases, i.e., a specific nucleotide sequence (*see* Fig. 4).

The apparent binding affinity of a TF to a particular DNA sequence is the sum of the nonspecific and the specific binding:

Affinity = Sum (general contacts) + Sum (specific contacts)

This shows that specific contacts only account for part of the binding affinity and that sequence variation has a quantitative effect on the binding affinity of TFs.

However, proteins as well as DNA are three-dimensional structures. There are sterical dependencies for each of the individual contacts. For example, a bulky side chain can only form a very limited number of sterical arrangements with respect to the DNA. This has direct consequences for the selection of neighboring positions in the DNA sequence. Due to such restrictions individual exchanges of nucleotides can have dramatically different effects on the binding affinity. This is not a simple linear function of sequence fit. The contribution of a DNA protein contact is a function of the sterical neighborhood and is easily represented by the nucleotide sequence. Taking this into consideration, the affinities are calculated as:

$$\text{general contact}_i = E_i * f(S_i) \quad \text{and} \quad \text{specific contact}_j = E_j * f(S_j)$$

where i and j are the position within the binding site, S_i is the sequence context of position i, and E_i is the energy of the interaction at position i.

The Characteristics of TF Binding Sites

TF binding sites are usually short 8–20 nucleotide stretches of DNA that are covered by the protein upon binding. Surprisingly, at first glance, the sequences for the binding sites for the same protein vary considerably in different locations. Nevertheless, the binding affinity may be comparable and all of these sites are part of a functional context. Interestingly, the average binding affinity of natural binding sites is often suboptimal, that is, there are other sequences that would bind with a much higher affinity to the protein. There are several reasons for these very general observations.

First, TFs have to bind to DNA during a lifetime of an individual or on a larger scale, until evolution has changed that particular regulatory event. Therefore, TF binding is faced with the inevitable mutations that occur over time in genomic DNA. By allowing a variety of slightly different sequences to be bound, TFs can reduce the risk of losing their ability for DNA binding due to point mutations in binding sites. Second, with regard to TF function, they not only need to specifically bind to DNA, but they must dissociate correctly in order to complete each transcriptional cycle.

The observed flexibility of TF binding sites has quite favorable consequences for biological function and fitness. However, how is this limited flexibility achieved on a molecular basis? This is important in order to design optimal algorithms for the detection of real binding sites by *in silico* methods. Detailed analysis of individual TF binding sites, binding the same factor, revealed that sequence variations are not evenly distributed over the binding site sequence. On one hand, there are a few highly conserved nucleotide positions within the binding site that are never or rarely changed. Mutations at such sites often have deleterious effects on DNA binding of the protein and have been shown in many cases to represent positions that are contacted by the protein in a sequence-specific manner. On the other hand, many binding sites contain positions that are covered by the protein as evident from footprint studies, but where there is low or no sequence conservation. Those *internal* spacers are usually not contacted in a base-specific manner and may or may not contribute back-bone contacts to binding (*see* Fig. 5).

The series of three-dimensional interactions can impart a mutual dependency of pairs of nucleotides within a binding site. This is evidenced as correlated positions.

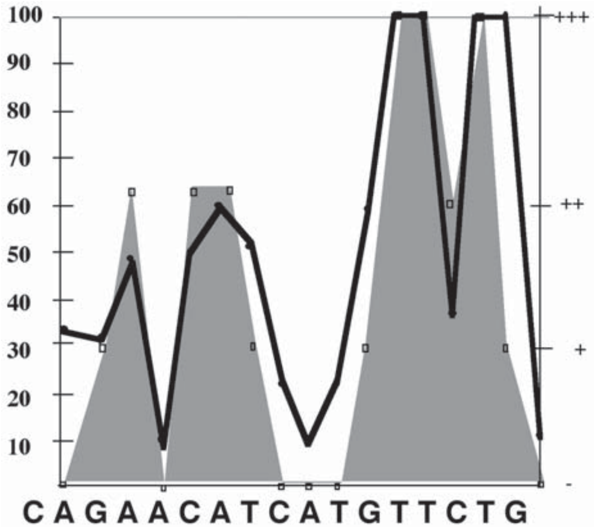


Fig. 5. Comparison of calculated importance (consensus index based on Shannon entropy) and experimental evidence for the Glucocorticoid Receptor Binding site. The scale on the left side shows the normalized consensus index derived from the nucleotide weight matrix of the GRE. The experimental evidence is given as follows: – = no DNA contact by the protein; + = backbone contact; ++ = unspecific base contact; +++ = sequence specific base contact.

These are different from conserved positions because correlated positions do vary in the nucleotide from sequence to sequence but in a coordinated manner (e.g., if position A contains an A then position B contains a G, if position A contains a T then position B contains an A). Thus, TF binding sites can be viewed as collections of closely related but distinct sequences that can be described in a collective manner. Thus, TF binding sites sequences only carry the potential to bind their cognate protein. They occur everywhere in the genome and are by no means restricted to regulatory regions. This complicates their recognition because there are always a large number of potential binding sites that have absolutely no function in transcriptional control. It is therefore almost impossible to discriminate functional from nonfunctional sites (usually referred to as *false positive matches*) based solely on the analysis of the individual binding sites.

The Principles of *In Silico* TF Binding Site Detection

The question becomes how does one identify these TF binding sites in nucleotide sequences? The first and obvious approach would be to collect known TF binding sites and then simply use a string matching algorithm to find new locations of the series of letters in question. Allowing for one or two mismatches to the known sites could mimic the sequence variations. As intriguing as this simple approach appears, it is so flawed in its practical application that it can only serve as a negative example. First, it can only identify binding sites already known if no mismatches are allowed. When mismatches are permitted this algorithm will return an overwhelming number of sequence matches, most of which may not even be binding sites for the factor in question.

A = A	**C** = C	**G** = G	**T** = T (U)
W = A or T	**S** = C or G	**K** = G or T	
M = A or C	**Y** = C or T		
R = A or G			
B = C, G, T **= (non A)**	**D** = A, G, T **= (non C)**	**H** = A, C, T **= (non G)**	**V** = A, C, G **= (non T)**
N = A, C, G,			

Fig. 6. The 15 letter IUPAC ambiguity code.

This illustrates the problem of this approach, ignoring the well-known variety of binding sites. Rather than using individual sequences, one can compile a consensus from all known binding sites and then search with this consensus to eliminate this problem. This is accomplished by aligning the binding sites by optimizing the number of matching positions then transforming this alignment into a single description. This can be accomplished by employing the International Union of Pure and Applied Chemistry (IUPAC) code that is an extended alphabet of nucleotides of 15 letters including symbols for alternative nucleotides (R = A or G). The first successful algorithms to locate TF binding sites in DNA have used this IUPAC string representation and they are still in use (*see* Fig. 6).

These approaches have limitations. First the IUPAC strings are absolutely unforgiving for sequences that were not part of the training set, i.e., the sequences used to derive the IUPAC consensus string. For example, if all training sequences contained either an A or a G at a particular position, a C at this position would cause all candidates to be rejected, even if they were identical to the IUPAC in all other positions. This may be correct in case of a sequence-specific recognition at this position but is unacceptable if this position is located within an internal spacer. IUPAC string matching yields a binary decision only; match or no match. There is no way to quantitatively distinguish between the candidates. In addition, the use of the IUPAC strings allows sequences to be identified that would not contain a functional binding site because the combinations of individual positions were not restricted by any means (allowing the accumulation of *rare* nucleotides throughout the binding site). Currently, the most reliable way to detect TF binding sites in DNA sequences is based on nucleotide weight matrices.

The Nucleotide Weight Matrix

The concept of nucleotide weight matrix (NWM) descriptions were developed in the 1980s. A weight matrix uses the complete composition of nucleotides for each position of the alignment to achieve a more differentiated rating of a matching sequence. For example, a single position of an alignment of 12 sequences containing (T,T,T,T,T,T,T,A,A,A,C,C; each letter representing one sequence at this position) would be assigned T in the IUPAC consensus. A new sequence with a T at this position would be considered a match while an A at the same place would cause the sequence to be dismissed as no match. Even a simple nucleotide distribution matrix would assign a weighted score (in this case proportional to the percentage of the nucleotide) of 0.58 to the T and still 0.25 to an A. In this manner, weight scores represent

Table 1
Internet Accessible Methods to Detect Promoter Elements (Transcription Factor Binding Sites)

Program	*Availability*	*Comments*
MatInspector	http://www.genomatix.de/cgi-bin/matinspector/matsearch.pl	MatInspector matrix library (includes matinspector.pl TRANSFAC matrices)
SIGNAL SCAN	http://wwwbimas.dcrt.nih.gov/molbio/signal/	IUPAC consensus library based on TFD
MATRIX SEARCH	http://wwwbimas.dcrt.nih.gov/molbio/matrixs/	IMD matrix library (TRANSFAC+TFD)
TFSearch	http://pdap1.trc.rwcp.or.jp/research/db/TFSEARCH.html	TRANSFAC matrices
TESS	http://agave.humgen.upenn.edu/utess/tess/	TRANSFAC Matrices

the similarity of the tested sequence to all of the sequences in the alignment much better than IUPAC consensus sequences. Most weight matrix-based methods use additional weighting. This is achieved by comparison of the actual nucleotide distribution with random values (log odds methods) or by other statistical measures. (e.g., information content).

The widespread use of weight matrices was delayed for almost a decade because only a few special matrices had been defined. In 1995 two (overlapping) matrix libraries for TF sites were compiled for the first time and became widely available. They still represent the only libraries of their kind and are used in several bioinformatics tool sets. Some of the tools are available through the world-wide-web (WWW) (*see* Table 1).

Available methods for TF binding site definition and detection have been surveyed and an extensive comparison of their capabilities was also published (*see* Suggested Reading).

The Hidden Markov Model

Consider tossing coins. There are just two possibilities for the outcome: heads or tails. You may attempt to find out what the probability is that either side will show up during the next coin toss given the outcome of the last event. This would constitute a simple Markov chain. Now consider, there is somebody behind a curtain (so you cannot observe the actual action) tossing coins and telling you the outcome. Again you attempt to predict the outcome from the previous result. However, this time things are complicated by the fact that the person behind the curtain has several coins to choose from and will not tell you which one was tossed. You are only told the final result while you do not know the start condition (i.e., which coin was tossed). If the selection of the coins is a stochastic background process, which cannot be observed directly (hidden) then calculating the probabilities for the various outcome scenarios from the previous outcome involving such a hidden background process constitutes a Hidden Markov Model (**HMM**).

If the selection of actual sequences used for the alignment of binding sites is considered a stochastic background process, then the probability of the occurrence of

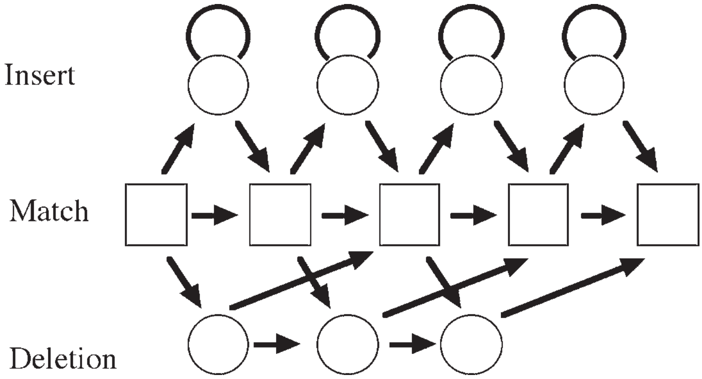


Fig. 7. Schematic architecture of a HMM describing the alignment of sequences with a length of five nucleotides. The square boxes indicate matching positions. The circles above the boxes indicate insertions and the circles below indicate deletions. Note that each sequence can feed through a different path in this HMM. The bold arrows represent the transitions and carry the parameters determining the relative probabilities for these transitions.

each of the four nucleotides at the next position of the alignment (of a new sequence to the existing alignment) can be calculated by a HMM. A schematic of a HMM is shown in Fig. 7.

As illustrated in Fig. 7., the parameters of the HMM are being trained from the existing alignment. In this manner any new candidate can be checked against the HMM describing a known binding site and quantitatively scored. HMMs have the advantage that they can account for almost all of the variability occurring in real sequences including the variable length of spacers within binding sites (not common but possible). Unfortunately, HMMs require a high number of parameters to describe binding sites with the required flexibility. Optimal training of all these parameters would require training sets of hundreds or even thousands of known binding sites for each factor. This is the sole reason why weight matrices still dominate the field of binding-site recognition although in theory HMMs are superior.

Frequent Problems with Practical Application of Search Programs

All search programs attempting to locate potential TF binding sites in genomic DNA face similar challenges. The most notorious, is incomplete data that prevents the generation of high quality descriptions by any means. For example, it is meaningless (but not impossible) to construct a weight matrix if only two examples for a specific binding site are available. Even a simpler IUPAC string will not be very useful. This is very close to the next most frequent problem, i.e., the *nonexistent description*. One must remember that if an IUPAC or matrix that describes a binding site is not available, it does not mean that a specific binding site for that protein does not exist.

Now, let us consider the case where there is a sufficiently well-defined weight matrix and this is used to scan sequences. Regardless of the matrix there will always be a disappointingly large number of matches in long sequences. Most of these matches will have no function in transcriptional control. The majority of the same matches may bind the cognate protein. The dilemma results not from the software, but from the confusion of two different topics. Transcriptional function only arises from TF binding sites located within the correct functional context. Accordingly, examination of the individual binding sites can never yield information about transcriptional function. This may seem trivial after the previous introduction. However, this incorrect expectation is very common among researchers and accounts for at least 95% of all concerns raised about TF binding site-detection programs.

The question then becomes, why are nonfunctional candidate sites identified that are obviously out of context? This can be addressed by considering that DNA sequences are used to encode many functions beyond transcription factor binding. For examples, the AP-1 binding site (IUPAC consensus TGASTCA) is several fold overrepresented in coding regions as compared to promoter regions. This puzzling observation has a very simple explanation. This combination of nucleotides encodes a series of frequently used amino acids. Occurrence of the AP-1 binding site is therefore a consequence of the coding potential of these sequences and has absolutely nothing to do with transcriptional regulation. However, there is no rule without an exception. There are coding regions that contain functional AP-1 binding sites.

How to Define Unknown Transcription Factor Binding Sites

In light of an ever-growing number of full-scale genomic sequencing projects it is very important to discover new, previously unknown binding sites. However, the question becomes how can a weight matrix be created for an unknown binding site when it requires a set of known binding sites?

It is difficult to identify transcription factor binding sites in a large totally anonymous sequence. However, there are several, at least partial ways out of this dilemma. Some previous level of knowledge is required to use pattern definition algorithms to produce new patterns that can be used as IUPAC consensus sequences or nucleotide weight matrices. A very effective and relatively simple approach is experimental determination of the binding site spectrum for a given protein. This is called *SELEX* that is actually the in vitro selection of binding sequences from a large collection of random oligonucleotides. Only sequences with sufficient affinity to the protein will be bound. The rest can be washed away. Sequencing of the bound oligonucleotides reveals individual binding sequences that can be used to derive a nucleotide weight matrix. However, before that can be done an extensive purification of the protein has to be carried out.

Another experimentally oriented approach is the evaluation of expression array data. Here, large amounts of gene probes are arrayed onto a filter or glass chip. This array is then hybridized to RNA (via complementary DNA) isolated from cells that have undergone some treatment. The amount of signal over each spot indicates the approximate level of RNA present for this gene. By comparing such values with a parallel experiment, e.g., with untreated cells, it is possible to detect which genes changed their RNA levels under treatment. Is is then possible to cluster genes according to their expression patterns. Analyzing promoter regions from the corresponding genes with a very similar expression pattern can be used to identify common patterns, many of which are transcription factor binding sites. Brazma, et al. (1998) have demonstrated this approach successfully in the yeast system. Unfortunately, promoters in higher eukaryotes, like mammals, are not as readily available as in yeast, but this will change in the near future. Regardless of the means by which promoter sequences are acquired, the sequences need to be analyzed for unknown motifs hidden in the set. There are many programs available to define the patterns in a set of sequences. The most popular methods are the Gibb's sampler reviewed in expectation maximization algorithms. A variety of other approaches are reviewed in Chapter 29 (*see* Suggested Reading).

Some Rules of Thumb for Application

It is most important to clearly define which kind of analysis makes sense and which does not. In contrast, the particular algorithm being used to detect TF binding sites is not as important. The rules of application are outlined below:

1. The first criterion is the length of the sequences to be analyzed. A binding site that appears to be a significant finding in a hundred base pairs may turn out to be matching every few hundred base pairs even in long random sequences. The smaller region selected for analysis (this can be localized, e.g., by experimental results or annotation), the more informative are results from TF binding site analysis.
2. Another important factor is the threshold to be used for the analysis. A very high threshold will most likely prevent the occurrence of false matches (those not binding to the protein). However, it will also suppress some true matches that just do not reach the high threshold. Optimal binding may not be the biological optimum, as dissociation must also occur at some time.
3. Using a low threshold will faithfully record all positive matches but will also collect a huge number of false matches. There are two solutions to this problem. First , observe the limitations of the sequence length: If it is not possible to restrict the length of the sequence to less than one kb then in most cases a low threshold will return substantially more data than is useful. One could look for binding sites within a larger context (at least another binding site near by), which allows for both lower thresholds as well as analysis of larger sequences.
4. If it is necessary to identify individual binding sites with low thresholds, sequence length should be 500 bp or less. It is worthwhile to concentrate one's efforts on meeting such conditions before carrying out the actual analysis for TF binding sites. Otherwise the results are usually not interpretable and the whole endeavor becomes a waste of time.

Nevertheless, even when observing the aforementioned conditions, results often contain more matches than desired. Evaluation of the matches is then required to select the most likely candidates for further analyses. Match scores are of limited use since scores do not necessarily reflect the binding affinities correctly and *weak* binding sites may be perfectly suitable within the correct context. Biological evidence should be used to evaluate the matches. Examples are correlation with other features (e.g., TSS, the closer the more likely involved in the promoter), conserved occurrence of sites in orthologous sequences in other species (indicative of functional conservation), or correlation with another binding site in promoters of coexpressed genes (indicative of promoter modules).

An example would be the development of a promoter model describing the histone H1 genes. In order to generate this model the following strategy was used: collecting suitable promoters for a set of histone H1 genes. We used the homology group 6.1.2.1.1 from the Eukeryotic Promoter Database (EPD) yielding the promoter sequences as seen in Table 2.

Table 2
Promoter Sequences from EPD Homology Group 6.1.2.1.1.

EPD AC	*EMBL AC (ID)*	*Description*
EP27002	M17019 (GGH1PL)	Gg histone H1-a 11L
EP27003	M17020 (GGH1PR)	Gg histone H1-a` 11R
EP27004	M17021 (GGH103)	Gg histone H1-b 03
EP27005	M17018 (GGH1LH)	Gg histone H1-c .10
EP07047	U18782 (GG18782)	Gg histone H1-c` 01
	J00863 (GGH11A1)	
EP33001	X06128 (APHISH1)	Cm histone H1
EP11065	X01752 (GGHISH1)	Gg histone H1-d 02
EP48019	Y12292 (MMHISTH14)	Mm histone H1-E
	L26163 (MMH1EH2B)	

Table 3
H1t and H1a Alignment Carried Out by CLUSTAL with Default Parameters

```
The following sequences are too divergent to be aligned:
hs_h1a and hs_h1t (distance 1.637)
(All distances should be between 0.0 and 1.0)
This may not be fatal but you have been warned!

hs_h1t         --GAAAATC-GAGGGCTTTCTCGAATAGTTTTGGCATCCAGGGT-CATTTTTCATTAAAA
hs_h1a         ATGAAAAGCTGAAGGGATTTTTTAAAATATCTTTCATCAATTGCACAAGATTCTTGAAAA
                 ***** * ** **  ** *  ** *  * *  **** *  *  **   *** * ****

hs_h1t         -AGAGAAAAGTCATGTCAAATATGAATTTCCGCAGATTATTCAGCACTAGACCCTGGGAG
hs_h1a         CACAAACAAGT-ATGTGAACCTGGAGGCTGTTTTCCTCCTTTGGAGCTTCAAAGTGCCAA
                * * * **** **** **    **   *       *  **  *  **  *   **  * 

hs_h1t         ATTCTGTAAAGAGGGGTTTTGT-TATACTCAACTTTTCCGGGTAAAACAAACACAAAT--
hs_h1a         ATTCTGTACCATTGTTTTAAGCATTTAATCAAATTTTGAGGACTAA-CAAACACAATTTG
               ********     *  **  *  * ** **** ****  **   ** ********* *  

hs_h1t         --ACTCCTCCTCCAAGG-GGCGG-----GGGCGGTGCCTAGGT-GATGCACCAATCACAG
hs_h1a         GGAGTCCAACGCGAGCGCGGCGGCCAGAGGGCGGTGGATTGGACGCTCCACCAATCACAG
                 * ***  * * *  * *****     ********  * **  * * ************

hs_h1t         CGC-GCCCTACCCTATATAAGGCCCCGAGGCCGCCCGGGTGTTTCATGCTTTTCGCTGGT
hs_h1a         GGCAGCGCCGGCTTATATAAG--CCCGGGCCCGAGCATAGCAGCAACGCAAAAC-CTGCT
                ** ** *   * ********  **** * ***  *         * **    * *** *

hs_h1t         TATTACATCTTGCGTTTCT-CTGTTGTTATGTCTGAAACCGTGCCTGCAGCTTCTGCCAG
hs_h1a         CTTTAGATTTCGAGCTTATTCTCTTCTAGCAGTTTCTTGCCACCATGNNNNNNNNNNNNN
                 *** ** * * * ** * ** ** *      *     *   * **             

hs_h1t         TGCTGGTGTAGCCGCTATGGAGAAAC
hs_h1a         NNNNNNNNNNNNNNNNNNNNNNNNNN
```

The term *homology group* is a bit misleading as the sequences only share sequence similarity around the TSS (by defintion position 500 in EPD promoters). This can be illustrated by an alignment of the best conserved part of two human histone H1 promoters, H1t and H1a (carried out by CLUSTAL with default parameters) as shown in Table 3.

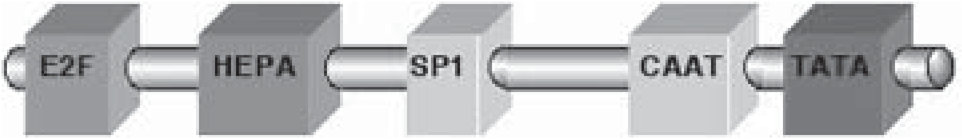


Fig. 8. Promoter Model.

Table 4
Transcription Elements Conserved in Histone H1 Promoters

Element	*Name*	*Strand*	*Parameters*	*Dist. to next element*
Matrix	V$E2FF	(+/–)	Min. core sim.: 0.750 Min. matrix sim.: optimized-0.07	0 to 75 bp
Matrix	V$HEPA	(–)	Min. core sim.: 0.800 Min. matrix sim.: optimized-0.07	10 to 100 bp
Matrix	V$SP1F	(+)	Min. core sim.: 0.750 Min. matrix sim.: 0.800	5 to 50 bp
Matrix	V$PCAT/ CAAT_01	(+)	Min. core sim.: 0.800 Min. matrix sim.: 0.860	20 to 30 bp
Matrix	V$TBPF	(+)	Min. core sim.: 0.800 Min. matrix sim.: 0.890	
				Total length: 35 – 255 bp

As clearly evident from the alignment there is some limited similarity that may be insufficient for recognition of the promoters. However, as pointed out, promoter function is encoded by a particular arrangement of transcription elements, in this case transcription factor binding sites. The next step is to format all of these promoter sequences for use as a training set, into one FASTA file and to analyze this file with GEMS Launcher (Genomatix software GmbH, Munich, Germany) for a conserved framework of binding sites.

See companion CD for color Fig. 8

This analysis results in a closely related framework, which can be summarized into the promoter model (*see* Fig. 8). This model consists of five binding sites in a linear arrangement with the characteristics described in Table 4.

It is interesting to note that E2F is known to be involved in cell-cycle regulation while the presence of a hepatic factor (HEPA) does not make immediate sense. Subsequently, one evaluates the specificity of the new model. For that purpose database searches were carried out using the EMBL database (*see Table 5*)

The complete results included 59 matches. The list shows the 30 histone gene promoters found by the model in one search. Most of the matches that were not shown were located in unannotated sequences. This shows that the model is quite specific (about 1 match every 12 million basepairs) and indeed locates new histone genes that were not in the training set (indicating that specificity is not just a trivial consequence of overtraining). As a negative control we purged EPD for histone promoters, result-

Table 5
Sequences Matched by Histone Promoter Model

Sequence Identifier [acc. #]	*Sequence Name*
MMHISH1T [M97756]	Macaca mulatta histone H1t gene
APHISH1 [X06128]	Duck histone H1 gene
GG18782 [U18782]	Gallus gallus histone H1 gene
GGH103 [M17021]	Chicken H1 histone gene lambda-Ch03 H1
GGH11A1 [J00863]	Chicken histone H1 gene
GGH1LH [M17018]	Chicken H1 histone gene lambda-H1.10
GGH1PL [M17019]	Chicken H1 histone gene pCH11.5E left
GGH1PR [M17020]	Chicken H1 histone gene pCH11.5E right
GGHISH1 [X01752]	Chicken histone H1 gene
-MM06232 [U06232]	Mus musculus CD-1 histone H1t (H1t) gene
MMH1EH2B [L26163]	Mouse histone H1e gene
-MMH1T [X72805]	M.musculus gene for testicular histone H1
-MMH1X [L26164]	Mouse histone H1
MMHIS1 [J03482]	Mouse histone H1 gene
-MMHIST431 [Y12290]	M.musculus genes encoding histone H1.1
-MMHISTA [L28753]	Mus musculus histone H1t (Hist) gene
MMHISTH12 [Y12291]	M.musculus genes encoding histone H1.2
MMHISTH14 [Y12292]	M.musculus genes encoding histone H1.4
MMU62922 [U62922]	Mus musculus histone H1b gene
RNH1D2B [X67320]	R.norvegicus genes for H1d-histone
RNH1TH4T [M28409]	Rat testis-specific histone H1t
RNHIS1D [M31229]	Rat histone H1d gene
RNHIS1TP [M13170]	Rat testis-specific H1 histone variant H1t gene
HS193B12 [Z98744]	Human histone H1.5 gene
HSH1FNC1 [X06757]	Human H1 histone gene FNC16 promoter region
HSH1T [M60094]	Human testicular H1 histone (H1) gene
HSHISAC [M60748]	Human histone H1 (H1F4) gene
HSHISH1T [M97755]	Human histone H1T gene
HSHISTN15 [X83509]	H.sapiens gene for histone H1.5.
HSU91328 [U91328]	Human histone 2A-like protein gene

ing in a total of 808,200 bp of 1347 non-histone H1 but functional promoters. The model had no false positive matches in this control set further supporting the extraordinary specificity.

Summary

Transcription factor binding sites are important elements in transcriptional control. TF binding sites per se only have the function of binding their cognate protein(s). The function in transcription control only arises within the appropriate context. TF binding sites represent a collection of variable sequences and are currently best described by nucleotide weight matrices. Since potential TF binding sites are not restricted to regulatory DNA regions, matches to search programs are best evaluated by looking at the functional context of candidates.

Glossary and Abbreviations

For a complete list of terms, *see* Website: http://www.genomatix.de/genomics_tutorials/basic_terms/basic_terms.html.

Enhancer DNA regions which are usually rich in *transcription factor binding sites* and/or repeats. They enhance transcription of the responsive *promoter* independent of orientation and position.

Expression Array Assessment of the expression of a gene in various cells/tissues or under specific conditions (stimulation by signals, development or differentiation) using *DNA-chips* and labeled hybridization probes.

Gene An entity of a genome consisting of sequences defining the gene product (RNA or protein) and additional sequences directing expression of the gene (regulatory sequences for transcriptional and post-transcriptional control). The human genome contains about 60,000 ± 30,000 genes.

Promoters DNA regions which are rich in *transcription factor binding sites* similar to enhancers, but also contain elements that allow specific initiation of transcription (core promoter).

Transcription Factor Binding Site Short stretches of DNA, sufficiently conserved to allow specific recognition by the corresponding transcription factor.

Suggested Readings

The Promoter

Sauer, F. and Tjian, R. (1997) Mechanisms of transcriptional activation: Differences and similarities between yeast, Drosophila, and man, Curr. Op. Gen. Dev. 7, 176–181.

Stamatoyannopoulos, J. A., Clegg, C. H., and Li, Q. (1997) Sheltering of gamma-globin expression from position effects requires both an upstream locus control region and a regulatory element 3' to the (A)gamma-globin gene, Mol. Cell. Biol. 17, 240–247.

The Characteristics of TF Binding Sites

Frech, K., Herrmann, G., and Werner, T. (1993) Computer-Assisted prediction, classification, and delimitation of protein binding sites in nucleic acids, Nucleic Acids Res. 21, 1655–1664.

Quandt, K., Frech, K., Karas, H., Wingender, E., and Werner, T. (1995) Matlnd and Matlnspector: New fast and versatile tools for detection of consensus matches in nucleotide sequence data, Nucleic Acids Res. 23, 4878–4884.

Risse, G., Jooss, K., Neuberg, M., Brüller, H.-J., Müller, R. (1989) Asymmetrical recognition of the palindromic AP1 binding site (TRE) by Fos protein complexes, EMBO J. 8, 3825–3832.

The Nucleotide Weight Matrix

Bucher, P. (1990) Weight matrix description of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences, J. Mol. Biol. 212, 563–578.

Chen, Q. K., Hertz, G. Z., and Stormo, G. D. (1995) Matrix Search 1.0: A computer program that scans DNA sequences for transcriptional elements using a database of weight matrices, Comp. Appl. Biosci. 11, 563–566.

Frech, K., Quandt, K., and Werner, T. (1997) Finding protein-binding sites in DNA sequences: The next generation, Trends Biochem. Sci. 22, 103–104.

Frech, K., Quandt, K., and Werner, T. (1997) Software for the analysis of DNA sequence elements of transcription, Comp. Appl. Biosci. 13, 89–97.

Ghosh, D. (1993) Status of the transcription factors database (TFD), Nucleic Acids Res. 21, 3117–3118.

Heinemeyer ,T., Wingender, E., Reuter, I., Hermjakob, H., Kel, A.E., Kel, O. V., et al. (1998) Databases on transcriptional regulation: TRANSFAC, TRRD and COMPEL, Nucleic Acids Res. 26, 362–367.

Naitou, M., Hagiwara, H., Hanaoka, F., Eki, T., and Murakami, Y. (1997) Expression profiles of transcripts from 126 open reading frames in the entire chromosome VI of Saccharomyces cerevisiae by systematic northern analyses, Yeast 13, 1275–1290.

Nagase, T., Ishikawa, K., Miyajima, N., Tanaka, A., Kotani, H., Nomura, N., and Ohara, O. (1998) Prediction of the coding sequences of unidentified human genes. IX. The complete sequences of 100 new cDNA clones from brain, which can code for large proteins in vitro, DNA Res. 5, 31–39.

Quandt, K., Frech, K., Karas, H., Wingender, E., and Werner, T. (1995) Matlnd and Matlnspector: New fast and versatile tools for detection of consensus matches in nucleotide sequence data, Nucleic Acids Res. 23, 4878–4884.

Stormo, G. D. and Hartzell, III, G. W. (1989) Identifying protein-binding sites from unaligned DNA fragments, Proc. Natl. Acad. Sci. USA 86, 1183–1187.

How to Define Unknown Transcription Factor Binding Sites

Brazma, A., Jonassen, I., Vilo, J., and Ukkonen, E. (1998) Predicting gene regulatory elements in silico on a genomic scale. Genome Res. 8, 1202–1215.

Cardon, L. R. and Stormo, G. D. (1992). Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments, J. Mol. Biol. 223, 159–170.

Neuwald, A. F., Liu, J. S., and Lawrence, C. E. (1995). Gibbs motif sampling: Detection of bacterial outer membrane protein repeats, Prot. Sci. 4, 1618–1632.

Scherf, M., Klingenhoff, A., and Werner, T. (2000) Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: A Novel Context Analysis Approach, J. Mol. Biol. 297, 599–606.

Wolfertstetter, F., Frech, K., Herrmann, G., and Werner, T. (1996) Identification of functional elements in unaligned nucleic acid sequences by a novel tuple search algorithm, Comp. Appl. Biosci. 12, 71–80.

31 An Introduction to Multiple Sequence Alignment and Analysis

Steven M. Thompson

Introduction

Given the nucleotide or amino acid sequence of a biological molecule, what do we know about that molecule? We can find biologically relevant information in sequences by searching for particular patterns that may reflect some function of the molecule. These can be catalogued motifs and domains, secondary structure predictions, physical attributes such as hydrophobicity, or even the content of DNA itself as in some of the gene-finding techniques. What about comparisons with other sequences? Can we learn about one molecule by comparing it to another? Yes, naturally we can; inference through similarity is fundamental to all the biological sciences. We can learn a tremendous amount by comparing our sequence against others.

The comparative method is a cornerstone of the biological sciences. Multiple sequence alignment is the comparative method on a molecular scale and is a vital prerequisite to some of the most powerful biocomputing techniques available. Many methods are available for aligning more than two sequences. Understanding the algorithms and the program parameters of each is the only way to rationally know what is appropriate. Several methods are even available on the Internet over www servers. Knowing and staying well within the limitations of this route will avert much frustration. However, realizing these limitations and being able to do something about them are very different things.

The power and sensitivity of sequence based computational methods dramatically increases with the addition of more data. More data yields stronger analyses if carefully carried out! Otherwise, it can confound the issue. The patterns of conservation become clearer by comparing the conserved portions of sequences among a larger and larger dataset. As in pairwise comparisons and database searching, those areas most resistant to change are functionally the most important to the molecule. The basic assumption is that those portions of sequence of crucial functional value are most constrained against evolutionary change. They will not tolerate many mutations. Not that mutations do not occur in these portions, just that most mutations in the region are lethal and thus not observed. Other areas of sequence are able to drift more readily being less subject to evolutionary pressure. Therefore, sequences become a mosaic of quickly and slowly changing regions over evolutionary time. We can use those constrained portions as *anchors* to create a sequence alignment allowing comparison. It is easy to see that two sequences are aligned when they have identical symbols at identical positions, but what happens when symbols are not identical or the sequences are not the same length? How can we know that the most similar portions of the sequences are aligned, when is an alignment optimal and does optimal mean biologically correct? Part of the solution is the dynamic programming algorithm.

Dynamic Programming

Let's review pairwise dynamic programming. Follow the example below. We will consider matching symbols to be worth one point, nonmatching symbols to be worth zero points, and we will use a very simple gap-penalty function. We will penalize the scoring scheme by subtracting one point for every gap inserted unless they are at the beginning or end of the sequence. In other words, end gaps will not be penalized, i.e., both sequences do not have to begin or end at the same point in the alignment. This zero penalty end-weighting scheme is the default for most alignment programs, but can often be changed with a program option. However, the gap function described here and used in the example below is a much simpler gap penalty function than used in most alignment programs. Normally an *affine*, i.e., a linear, function is used; the standard $y = mx + b$ equation:

total penalty = gap opening penalty + [(length of gap) · (gap extension penalty)]

To execute most alignment programs with this simple DNA gap penalty, you have to designate a gap *creation* or *opening* penalty of zero and a gap *extension* penalty of whatever counts in that particular program as an identical base match for DNA sequences.

As we will see, the oversimplified gap function used in this example does have a rather unusual effect. The solution occurs in two stages. The first begins very much like a dot matrix method while the second is totally different. A simple representation is used here. Instead of calculating the *score matrix* on the fly as you proceed through the graph, as is often done in illustrations of dynamic programming, I like to completely fill in an original *match matrix* first, and then add points to those positions that produce favorable alignments next. Points are added based on a "*looking back over-your-left-shoulder*" algorithm rule where the only allowable trace-back is diagonally behind and above. Follow the example in Table 1.

There will probably be more than one best path through the matrix. This time, starting at the top and working down as we did, then tracing back, I found two optimum alignments:

```
cTATAtAagg          cTATAtAagg
|  |||||            |   ||||
cg.TAtAaT.          cgT.AtAaT.
```

Each of these solutions yields a trace-back total score of 22. This is the number optimized by the algorithm, not any type of a similarity or identity score! Even though one of these alignments has six exact matches and the other has five, they are both optimal according to the rather strange gap-penalty criteria in which we solved the algorithm. Do you have any ideas about how others could be discovered? Often if you reverse the solution of the entire dynamic programming process, other solutions are found! In other words, reverse the sequences in software programs to see alternative alignments. Some programs also offer a highroad/lowroad option to help explore this solution space. To summarize, an optimal pairwise alignment is defined as an arrangement of two sequences, the first of length *i* and the second of length *j*, such that:

Table 1
Dynamic Programming Example[a]

a. First complete a match matrix using one point for matching and zero points for mismatching between bases:

	c	*T*	*A*	*T*	*A*	*t*	*A*	*a*	*g*	*g*
c	1	0	0	0	0	0	0	0	0	0
g	0	0	0	0	0	0	0	0	1	1
T	0	1	0	1	0	1	0	0	0	0
A	0	0	1	0	1	0	1	1	0	0
t	0	1	0	1	0	1	0	0	0	0
A	0	0	1	0	1	0	1	1	0	0
a	0	0	1	0	1	0	1	1	0	0
T	0	1	0	1	0	1	0	0	0	0

b. Now add and subtract points based on the best path through the matrix, working diagonally, left to right and top to bottom. When you have to jump a box to make the path, subtract one point per box jumped, except at the beginning or end of the alignment. Fill in all additions and subtractions and calculate the sums and differences as you go:

	c	*T*	*A*	*T*	*A*	*t*	*A*	*a*	*g*	*g*
c	1	0	0	0	0	0	0	0	0	0
g	0	0+1=1	0+1–1=0	0+0–0=0	0+0–0=0	0+0–0=0	0+0–0=0	0+0–0=0	1+0–0=1	1+0=1
T	0	1+1–1=1	0+1=1	1+1–1=1	0+0–0=0	1+0–0=1	0+0–0=0	0+0–0=0	0+0–0=0	0+1–0=1
A	0	0+0–0=0	1+1=2	0+1=1	1+1=2	0+1–1=0	1+1=2	1+1–1=1	0+0–0=0	0+0–0=0
t	0	1+0–0=1	0+1–1=0	1+2=3	0+1=1	1+2=3	0+2–1=1	0+2=2	0+1=1	0+0–0=0
A	0	0+0–0=0	1+1=2	0+2–1=1	1+3=4	0+3–1=2	1+3=4	1+3–1=3	0+2=2	0+1=1
a	0	0+0–0=0	1+0–0=1	0+2=2	1+3–1=3	0+4=4	1+4–1=4	1+4=5	0+3=3	0+2=2
T	0	1+0–0=1	0+0–0=0	1+1=2	0+2=2	1+3=4	0+4=4	0+4=4	0+5=5	0+5–1=4

c. Clean up the score matrix next. I'll only show the totals in each cell here:

	c	*T*	*A*	*T*	*A*	*t*	*A*	*a*	*g*	*g*
c	1	0	0	0	0	0	0	0	0	0
g	0	1	0	0	0	0	0	0	1	1
T	0	1	1	1	0	1	0	0	0	1
A	0	0	2	1	2	0	2	1	0	0
t	0	1	0	3	1	3	1	2	1	0
A	0	0	2	1	4	2	4	3	2	1
a	0	0	1	2	3	4	4	5	3	2
T	0	1	0	2	2	4	4	4	5	4

d. Finally, convert the score matrix into a trace-back path graph by picking the bottom-most, furthest right and highest scoring coordinates. Then choose the highest scoring trace-back route, to connect them all the way back to the beginning using the same 'over-your-left-shoulder' rule:

	c	*T*	*A*	*T*	*A*	*t*	*A*	*a*	*g*	*g*
c	①	0	0	0	0	0	0	0	0	0
g	0	①	0	0	0	0	0	0	1	1
T	0	1	①	①	0	1	0	0	0	1
A	0	0	2	1	②	0	2	1	0	0
t	0	1	0	3	1	③	1	2	1	0
A	0	0	2	1	4	2	④	3	2	1
a	0	0	1	2	3	4	4	⑤	3	2
T	0	1	0	2	2	4	4	4	⑤	4

[a] This example uses two randomly generated sequences that happen to fit the TATA consensus regions of eukaryotes and prokaryotes. The most conserved bases within the consensus are capitalized. The eukaryote promoter sequence is along the X-axis, the prokaryote along the Y-axis. Circled digits indicate trace-back route.

1. You maximize the number of matching symbols between 1 and 2;
2. You minimize the number of gaps within 1 and 2; and
3. You minimize the number of mismatched symbols between 1 and 2.

Therefore, the actual solution can be represented by:

$$S_{ij} = s_{ij} + \max \left\{ \begin{array}{ll} S_{i-1\;j-1} & \\ & \text{or} \\ \max S_{i-x\;j-1} + w_{x-1} & \text{or} \\ 2 < x < i & \\ \max S_{i-1\;j-y} + w_{y-1} & \text{or} \\ 2 < y < i & \end{array} \right\}$$

where S_{ij} is the score for the alignment ending at i in sequence 1 and j in sequence 2,
s_{ij} is the score for aligning i with j,
w_x is the score for making a x long gap in sequence 1,
w_y is the score for making a y long gap in sequence 2,

allowing gaps to be any length in either sequence.

Just because dynamic programming guarantees an optimal alignment, it is not necessarily the only optimal alignment. Furthermore, the optimal alignment is not necessarily the correct or biologically relevant alignment! As always, question the results of any computerized solution based on what you know about the biology of the system. The above example illustrates the Needleman and Wunsch (1970) global solution. Later refinements, like Smith and Waterman (1981), demonstrate how dynamic programming could also be used to find optimal local alignments. Programs use the following two means to solve dynamic programming using local alignments.

1. An identity match matrix that uses negative numbers for mismatches is incorporated. Therefore, bad paths quickly become very bad. This leads to a trace-back path matrix with many alternative paths, most of which do not extend the full length of the graph.
2. The best trace-back within the graph is chosen. This does not have to begin or end at the edges of the graph; it is looking for the best segment of the alignment!

Scoring Matrices

What about protein sequences, conservative replacements and similarities, as opposed to identities? This is definitely an additional complication to consider. Certain amino acids are very much alike, structurally, chemically, and genetically. How can we take advantage of the similarity of amino acids in our alignments? Margaret Dayhoff (1979) unambiguously aligned closely related protein datasets with no more than 15% difference available at that point in time. She noticed that certain residues, if they mutate at all, are prone to change into certain other residues. These propensities for change fell into the same categories that chemists had known for years, those chemical and structural classes, conserved through the evolutionary constraints of natural selection. However, Dayhoff's empirical observation quantified these changes. Based on the multiple sequence alignments that revealed a level of divergence between the sequences considered, the assumption that estimated mutation rates in closely related proteins can be extrapolated to more distant relationships was born. Using matrix and logarithmic mathematics that smooth out the statistics of the system,

Table 2
BLOSUM62 Scoring Matrix

	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V	W	X	Y	Z
A	4	–2	0	–2	–1	–2	0	–2	–1	–1	–1	–1	–2	–1	–1	–1	1	0	0	–3	–1	–2	–1
B	–2	6	–3	6	2	–3	–1	–1	–3	–1	–4	–3	1	–1	0	–2	0	–1	–3	–4	–1	–3	2
C	0	–3	9	–3	–4	–2	–3	–3	–1	–3	–1	–1	–3	–3	–3	–3	–1	–1	–1	–2	–1	–2	–4
D	–2	6	–3	6	2	–3	–1	–1	–3	–1	–4	–3	1	–1	0	–2	0	–1	–3	–4	–1	–3	2
E	–1	2	–4	2	5	–3	–2	0	–3	1	–3	–2	0	–1	2	0	0	–1	–2	–3	–1	–2	5
F	–2	–3	–2	–3	–3	6	–3	–1	0	–3	0	0	–3	–4	–3	–3	–2	–2	–1	1	–1	3	–3
G	0	–1	–3	–1	–2	–3	6	–2	–4	–2	–4	–3	0	–2	–2	–2	0	–2	–3	–2	–1	–3	–2
H	–2	–1	–3	–1	0	–1	–2	8	–3	–1	–3	–2	1	–2	0	0	–1	–2	–3	–2	–1	2	0
I	–1	–3	–1	–3	–3	0	–4	–3	4	–3	2	1	–3	–3	–3	–3	–2	–1	3	–3	–1	–1	–3
K	–1	–1	–3	–1	1	–3	–2	–1	–3	5	–2	–1	0	–1	1	2	0	–1	–2	–3	–1	–2	1
L	–1	–4	–1	–4	–3	0	–4	–3	2	–2	4	2	–3	–3	–2	–2	–2	–1	1	–2	–1	–1	–3
M	–1	–3	–1	–3	–2	0	–3	–2	1	–1	2	5	–2	–2	0	–1	–1	–1	1	–1	–1	–1	–2
N	–2	1	–3	1	0	–3	0	1	–3	0	–3	–2	6	–2	0	0	1	0	–3	–4	–1	–2	0
P	–1	–1	–3	–1	–1	–4	–2	–2	–3	–1	–3	–2	–2	7	–1	–2	–1	–1	–2	–4	–1	–3	–1
Q	–1	0	–3	0	2	–3	–2	0	–3	1	–2	0	0	–1	5	1	0	–1	–2	–2	–1	–1	2
R	–1	–2	–3	–2	0	–3	–2	0	–3	2	–2	–1	0	–2	1	5	–1	–1	–3	–3	–1	–2	0
S	1	0	–1	0	0	–2	0	–1	–2	0	–2	–1	1	–1	0	–1	4	1	–2	–3	–1	–2	0
T	0	–1	–1	–1	–1	–2	–2	–2	–1	–1	–1	–1	0	–1	–1	–1	1	5	0	–2	–1	–2	–1
V	0	–3	–1	–3	–2	–1	–3	–3	3	–2	1	1	–3	–2	–2	–3	–2	0	4	–3	–1	–1	–2
W	–3	–4	–2	–4	–3	1	–2	–2	–3	–3	–2	–1	–4	–4	–2	–3	–3	–2	–3	11	–1	2	–3
X	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1	–1
Y	–2	–3	–2	–3	–2	3	–3	2	–1	–2	–1	–1	–2	–3	–1	–2	–2	–2	–1	2	–1	7	–2
Z	–1	2	–4	2	5	–3	–2	0	–3	1	–3	–2	0	–1	2	0	0	–1	–2	–3	–1	–2	5

[a] Values whose magnitude is ± 4 are drawn in outline characters to make them easier to recognize. Notice that positive values for identity range from 4 to 11 and negative values for those substitutions that rarely occur are as low as –4. The most conserved residue is tryptophan with an identity score of 11; cysteine is next, with a score of 9; histidine, 8; both proline and tyrosine have scores of 7. The hydrophobic substitutions: isoleucine, leucine, valine, and to a lesser extent, methionine, easily swap places.

she was able to empirically specify the relative probabilities that different residues mutate into other residues through evolutionary history. This is the basis of the famous PAM (corrupted acronym of accepted point mutation) 250 (meaning that the matrix has been multiplied by itself 250 times) log odds matrix. Since Dayhoff's time, other biomathematicians, (*see* Henikoff and Henikoff's BLOSUM series of tables) have created newer matrices with more or less success than Dayhoff. Dayhoff's original PAM 250 table remains a classic as historically the most widely used.

Collectively these types of tables are known as symbol comparison tables, log odds matrices, or scoring matrices and they are fundamental to all sequence comparison techniques. The standard default scoring matrix for many protein similarity comparison programs is now the BLOSUM62 table. It appears in Table 2.

Rather than using the one/zero match function shown in the simple dynamic programming used for nucleic acids, protein sequence alignments use the match function provided by a scoring matrix. The concept of similarity becomes very important with some amino acids being *more similar* than others! A common misnomer in this area is the concept of homology vs similarity: there is a huge difference! Similarity is merely a statistical parameter that describes how two sequences, or portions of them, are alike according to some set scoring criteria. It can be normalized to ascertain statistical significance as seen in the database searching methods. Homology, in contrast and by definition, implies an evolutionary relationship. You need to be able to demonstrate lineage between the organisms or genes of interest in order to claim homology or provide experimental evidence, e.g., morphological, genetic, or fossil, that corroborates your assertion. The term *percent homology* does not exist; something is either homologous or it is not. The famous molecular evolutionist Walter Fitch likes to relate the joke, "homology is like pregnancy; you can't be 45% pregnant, just like something can't be 45% homologous. You either are or you are not." Do not make the all too commonly made mistake of calling any sequence similarity homology. Highly significant similarity can be used to argue homology, but never the other way around.

Multiple Sequence Dynamic Programming

As seen in pairwise dynamic programming, a *brute force* approach, looking at every possible position by sliding one sequence along every other sequence, is just not practical for alignment. Even without considering the introduction of gaps, the computation required to compare all possible alignments between just two sequences requires time proportional to the product of the lengths of the two sequences. Therefore, if the two sequences are approx the same length (N), this is a N^2 problem. To include gaps, the calculation would be repeated 2N times to examine the possibility of gaps at each position within the sequences. This is now a $N4^N$ problem! Dynamic programming reduces the problem to N^2.

How do you work with more than just two sequences at a time? You could painstakingly manually align all your sequences using some type of editor, and many people do that, but an automated solution is desirable, at least as a starting point to manual alignment. However, solving the dynamic programming algorithm for more than just two sequences rapidly becomes intractable. Dynamic programming's complexity, and hence its computational requirements, increases exponentially with the number of sequences in the dataset being compared [complexity = (sequence length)$^{\text{number of sequences}}$]. Mathematically this is an N-dimensional matrix, quite complex indeed. As we have seen, pairwise dynamic programming solves a two-dimensional matrix and the complexity of the solution is equal to the length of the longest sequence squared. A three-member standard dynamic programming sequence comparison would be a matrix with three axes, the length of the longest sequence cubed, and so forth. You can at least draw a three-dimensional matrix, but more than that becomes difficult, if not impossible, to even visualize. It quickly boggles the mind.

Several different heuristics have been employed over the years to simplify the complexity of the problem. One program, MSA, attempts to globally solve the N-dimensional matrix equation using a bounding box trick. However, the algorithm's complexity precludes its use in most situations, except with very small datasets. One way to globally solve the algorithm and yet reduce its complexity is to restrict the search space to only the most conserved *local* portions of all the sequences involved. This approach is used by the program PIMA. MSA and PIMA are both available through the Internet.

How the Algorithm Works

The most common implementations of automated multiple alignment modify dynamic programming by establishing a pairwise order in which to build the alignment. This modification is known as pairwise, progressive dynamic programming. Originally attributed to Feng and Doolittle (1987), this variation of the dynamic programming algorithm generates a global alignment, but restricts its search space at any one time to a local neighborhood of the full length of only two sequences. Consider a group of sequences. First, all pairs are compared to each other, using normal dynamic programming. This establishes an order for the set, most to least similar. Similarly, subgroups are clustered together. Then, take the top two most similar sequences and align them using normal dynamic programming. Now create a consensus of the two and align that consensus to the third sequence using standard dynamic programming. Now create a consensus of the first three sequences and align that to the forth most similar. This process continues until it has worked its way through all sequences and/or sets of clusters. The pairwise, progressive solution is implemented in several programs including Des Higgins' and Julie Thompson's Clustal (1994) (a copy of which is available on this volume's accompanying CD) and the GCG PileUp program.

See companion CD

As seen with pairwise alignments and sequence database similarity searching, all of this is much easier with protein sequences vs nucleotide sequences. Twenty symbols are easier to align then only four; the signal-to-noise ratio is far better. Furthermore, the concept of similarity applies to amino acids but generally not to nucleotides. If at all possible, multiple sequence alignment should always be carried out at the protein level. Therefore, translate nucleotide sequences to their protein counterparts if you are aligning coding sequences before performing multiple sequence alignment. The process is much more difficult if you are forced to align nucleotides because the region does not code for a protein. Automated methods may be able to provide a starting point, but there is no guarantee that the alignment will be biologically correct. The resulting alignment will probably require extensive editing, if it works at all. Success will largely depend on the similarity of the nucleotide dataset.

One liability of the global progressive pairwise methods is they are entirely dependent on the order in which the sequences are aligned. Fortunately, ordering them from most similar to least similar usually makes biological sense and works very well. However, the techniques are very sensitive to the substitution matrix and specified gap penalties. Programs that allow *fine-tuning* areas of an alignment by re-alignment with different scoring matrices and/or gap penalties can be extremely helpful. However, any automated multiple sequence alignment program should be thought of as only a tool to offer a starting alignment that can be improved upon, not the *end-all-to-meet-all* solution, guaranteed to provide the *one-true* answer.

Reliability

To help assure the reliability of the sequence alignment, always use comparative approaches. A multiple sequence alignment is a hypothesis of evolutionary history. To insure that you have prepared a reasonable alignment, be sure it makes sense. Think about it, a sequence alignment is a statement of positional homology. It establishes the explicit homologous correspondence of each individual sequence position, each column in the alignment. Therefore, devote considerable time and energy toward developing the most satisfying multiple sequence alignment possible. Editing alignments is encouraged. Specialized sequence editing software helps achieve this, but any editor will do as long as the sequences are properly formatted. After some automated solution has offered its best guess, go into the alignment and use your own brain to improve it. Use all available information and understanding to insure that all columns are homologous. Look for conserved functional sites to help guide your judgement. Assure that known enzymatic, regulatory, and structural elements all align. The results of subsequent analyses are absolutely dependent on the alignment.

Researchers have successfully used the conservation of co-varying sites in ribosomal and other structural RNA alignments to assist refinement. That is, as one base in a stem-structure changes the corresponding Watson-Crick paired base will change in a corresponding manner. The Ribosomal Database Project at the Center for Microbial Ecology at Michigan State University has used this process extensively to help guide the construction of their rRNA alignments and structures (*see* Website: http://rdp.cme.msu.edu/html/index.html).

Be sure an alignment makes biological sense. Beware of comparing *apples and oranges*. If creating alignments for phylogenetic inference, either make *paralogous* comparisons (i.e., evolution via gene duplication) to ascertain gene phylogenies within one organism, or *orthologous* (within one ancestral loci) comparisons to ascertain gene phylogenies between organisms, which should imply organismal phylogenies. Try not to mix them up without complete data representation. A substantial amount of confusion can arise, especially if you do not have all the data and/or if the nomenclature is contradictory; extremely misleading interpretations can result. Be wary of trying to align genomic sequences with cDNA when working with DNA, the introns will cause all sorts of headaches. Similarly, do not align mature and precursor proteins from the same organism and loci. It does not make evolutionary sense, as one has not evolved from the other, rather one is the other. These are all easy mistakes to make, try your best to avoid them.

Some general guidelines to remember include the following:

- If the homology of a region is in doubt, then throw it out (or *mask* it).
- Avoid the most diverged parts of molecules, they are the greatest source of systematic error.
- Do not include sequences that are more diverged than necessary for the analysis at hand.

Practical consideration: remember the old adage: "garbage in—garbage out!"

Applicability

The question arises, what are the uses of multiple sequence alignments? They are:

- Very useful in the development of PCR primers and hybridization probes;
- Great for producing annotated, publication quality, graphics and illustrations;
- Invaluable in structure/function studies through homology inference;
- Essential for building "Profiles" for remote homology similarity searching;
- Required for molecular evolutionary phylogenetic inference programs such as those from Phylogenetic Analysis Using Parsimony (and other methods) (PAUP*) and PHYLogeny Inference Package (PHYLIP).

The results from a multiple sequence alignment are useful for probe and primer design. They allow you to visualize the most conserved regions of an alignment. This technique is invaluable for designing phylogenetic specific probes as it clearly localizes areas of high conservation and high variability in an alignment. Depending on the dataset that you analyze, any level of phylogenetic specificity can be achieved. Areas of high variability in the dataset can be used to differentiate between universal and specific probe sequences. After localizing these general target areas, you can then use any of a number of primer discovery programs to find the best primers within those regions and to test those potential probes for common PCR conditions and problems.

Graphics prepared from multiple sequence alignments can dramatically illustrate functional and structural conservation. These can take many forms of all or portions of an alignment, shaded or colored boxes or letters for each residue, cartoon representations of features, running line graphs of overall similarity, overlays of attributes, various consensus representations. All can be printed with high-resolution equipment, in color or gray tones. These can make a big difference in a poster or manuscript presentation.

Conserved regions of an alignment are functionally important. In addition to the conservation of primary sequence and function, structure is also conserved in these crucial regions. In fact, recognizable structural conservation between true homologs extends way beyond statistically significant sequence similarity. An often-cited example is in the serine protease superfamily. *S. griseus* protease A demonstrates remarkably little similarity when compared to the rest of the superfamily (Expectation-values E 10 $^{1.8}$ in a typical search) yet its three-dimensional structure clearly shows its allegiance to the serine proteases (Pearson, W.R., personal communication). These principles are the premise of *homology modeling* and it works remarkably well.

As originally described by Michael Gribskov (1987), profiles are a position specific weight matrix description of an alignment or a portion of an alignment. Gap insertion is penalized more heavily in conserved areas than in variable regions, and the more highly conserved a residue is, the more important it becomes. Later refinements have added more statistical rigor. Generally, a profile is created from an alignment of related sequences and then used to search databases for remote sequence similarities. Profile searching is tremendously powerful and can provide the most sensitive, albeit extremely computationally intensive, database similarity searches possible.

We can use multiple sequence alignments to infer phylogeny. Based on the assertion of homologous positions in an alignment, several algorithms can estimate the most reasonable evolutionary tree for that alignment. Always remember that regardless of the algorithm used, parsimony, any distance method, or even maximum likelihood, all molecular sequence phylogenetic inference programs make the absolute validity of your input alignment their first and most critical assumption.

The most important factor to infer reliable phylogenies is the accuracy of the multiple sequence alignment. The interpretation of your results is utterly dependent on the quality of your input. In fact, many experts advice against using any parts of the sequence data that are at all questionable. Only analyze those portions that assuredly align. As a general rule, if any portions of the alignment are in doubt, throw them out. This usually means trimming down or masking the alignment's terminal ends and may require internal trimming or masking as well. Biocomputing is always a delicate balance—signal against noise—and sometimes it can be quite the balancing act.

Complications

One of the biggest problems in computational biology is that of sequence format. Each suite of programs seems to require its own different sequence format. The major databases all have their own. Clustal has its own; even the database similarity searching program FastA has a sequence format associated with it. The GCG Package sequence format exists both as single and *Multiple Sequence Format* (MSF) forms, and GCG's SeqLab has its own format called *Rich Sequence Format* (RSF) that contains both sequence data and reference and feature annotation. PAUP* has a required format called the *NEXUS* file and PHYLIP has its own unique input data format requirements. The PAUP* interfaces in the GCG Package, PAUPSearch, and PAUPDisplay, automatically generate their required NEXUS format directly from the GCG formatted files. Most systems are not nearly so helpful. Several different programs are available to convert formats back and forth between the required standards, but it all can get quite confusing. One program, ReadSeq by Don Gilbert at Indiana University (1993), allows for the back and forth conversion between several different formats. I would heartily recommend installing it on all of your computers. It is available as a C version or a new JAVA version with a graphical interface. Alignment gaps are another problem. Different program suites may use different symbols to represent them. Most programs use hyphens, "-", the GCG Package uses periods, ".". Furthermore, not all gaps in sequences should be interpreted as deletions. Interior gaps are probably okay to represent this way, as regardless of whether a deletion, insertion, or a duplication event created the gap; they will be treated the same by the algorithms. These are called *indels*. However, end gaps should not be represented as indels because a lack of information beyond the length of a given sequence may not be due to a deletion or insertion event. It may have nothing to do with the particular stretch being analyzed at all. It may just not have been sequenced. These gaps are just place holders for the sequence. Therefore, it is safest to manually edit an alignment to change leading and trailing gap symbols to +, *unknown amino acid,* or n, *unknown base*, or ?, which is supported by many programs and means *unknown residue or indel*. This will ensure that the programs do not make incorrect assumptions about your sequences.

The Protein System

The Elongation Factors are a vital protein family crucial to protein biosynthesis. They are ubiquitous to all of cellular life and, in concert with the ribosome, they must have been one of the very earliest enzymatic factories in life. The Elongation Factor subunit known as 1-Alpha (EF-1α) in Eukaryota and Archaea and called *Elongation Factor Tu* in [Eu]Bacteria (and Euk and Arch plastids) will be used as an example. It is essential in the universal process of protein biosynthesis and promotes the GTP-dependent binding of aminoacyl-tRNA to the A-site of the intact ribosome. GTP is hydrolyzed to GDP in the process. Because of strong evolutionary pressure resulting in very slow divergence and because of its ubiquity, it is an appropriate gene to estimate early life questions. In fact, a series of papers in the early 1990s, notably those by Iwabe, et al. (1989), Rivera and Lake (1992), and Hasegawa, et al. (1997) all base *universal* trees of life on this gene. Iwabe, et al. used the trick of aligning the α gene paralog EF-1β to their α dataset to root the tree. Elongation Factor 1α/Tu has guanine nucleotide, ribosome, and aminoacyl-tRNA binding sites. There are three distinct types of elongation factors that all work together to help perform the vital function of protein biosynthesis, as seen in Table 3.

Table 3
The Three Elongation Factors in Eukayota and Bacteria

Eukaryota	*[Eu]Bacteria*	*Function*
EF-1α	EF-Tu	Binds GTP and an aminoacyl-tRNA; delivers the latter to the A site of ribosomes.
EF-1β	EF-Ts	Interacts with EF-1α/EF-Tu to displace GDP and thus allows the regeneration of GTP-EF-1α/EF-Tu
EF-2	EF-G	Binds GTP and peptidyl-tRNA and translocates the latter from the A site to the P site.

[a] In [Eu]Bacteria and Eukaryota they have the names as described. The nomenclature in Archaea is still under consideration.

In EF-1α, a specific region is involved in a conformational change mediated by the hydrolysis of GTP to GDP. This region is conserved in both EF-1α/EF-Tu and EF-2/EF-G and seems to be typical of GTP-dependent proteins, which bind noninitiator tRNAs to the ribosome.

In *E. coli* EF-Tu is encoded by a duplicated loci; tufA and tufB. They are located at positions 74.92 and 90.02 of the chromosome. In humans at least 20 loci on seven different chromosomes demonstrate homology to this gene. However, only two are potentially active. The remainder appear to be retropseudogenes. The gene is encoded in both the nucleus and mitochondria and chloroplast genomes in eukaryotes and is a globular, cytoplasmic enzyme in all life forms.

The three-dimensional structure of elongation factor 1α/Tu has been solved in about 15 cases. Partial and complete *E. coli* structures have been resolved and deposited in the Protein Data Bank (1EFM, 1ETU, 1DG1, 1EFU, and 1EFC). The complete *Thermus aquaticus* and *thermophilus* structures have been determined (1TTT, 1EFT, and 1AIP), and even the cow EF-1α has been determined (1D2E). Most of the structures show the protein in complex with its nucleotide ligand, some show the terniary complex. The *T. aquaticus* structure is shown in Fig. 1. The *T. aquaticus* structure has six well-defined helices that occur from residue 24 through 38, 86 through 98, 114 through 126, 144 through 161, 175 through 184, and 194 through 207. There are also two short helices at residues 47 to 51 and 54 to 59. The guanine nucleotide-binding site involves the following regions: residues 18 to 25, residues 81 to 85, and residues 136 to 139. Residue 8 is associated with aminoacyl-tRNA binding. Multiple sequence alignment with this dataset can be used to explore these functional and structural regions as well as discover other interestingly conserved sites. To illustrate the various principles, we will restrict the example to a subset of *lower* eukaryotic EF-1α sequences. These will include many protists and algae but will exclude much of the *Crown* group, including all of the higher plants, true fungi, and metazoans. As such it may be an appropriate dataset with which to ask early branching order questions deep in eukaryotic evolution.

See companion CD for color Fig. 1

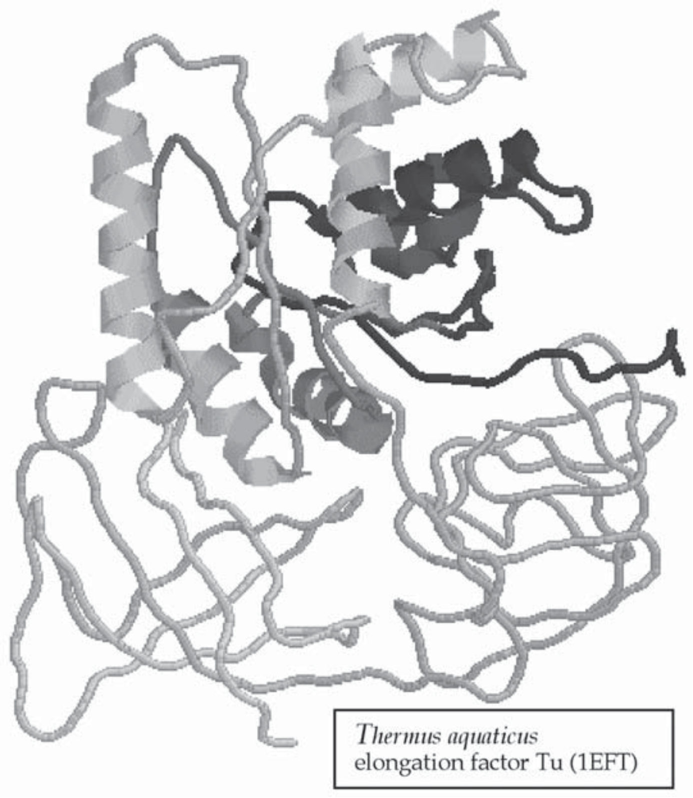


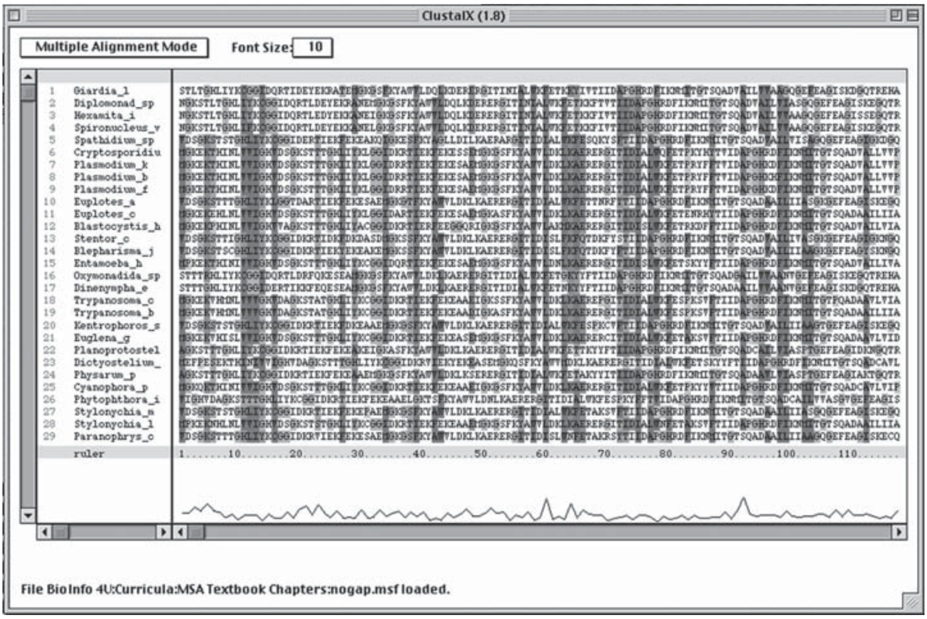
Fig. 1. The *Thermus aquaticus* elongation factor Tu structure, 1EFT. Notice that half of the protein has well-defined alpha helices and the rest consists of rather unordered coils. GTP/GDP fits right down in amongst all the helices in the pocket.

What is Available

A large range of programs is available for performing multiple sequence alignment on your own computer or on server computers. Many of the client/server applications are web-based so the only program requirement on your part is a Web browser. Specialized programs such as the global MSA and local PIMA software are generally not loaded onto personal machines, most are run over Web links. However, running a general purpose, progressive, pairwise implementation for multiple alignment directly on your own computer can often be very helpful. This is especially true if your Internet connection is slow or unreliable. Perhaps the most popular of the general purpose multiple sequence alignment programs is Clustal. The current version, ClustalW and its multi-platform, graphical user interface ClustalX can be found at biocomputing sites around the globe and on the CD accompanying this volume, and installed on your own machine. ClustalX has versions available for most graphical computer Operating Systems; including UNIX, Microsoft Windows, and Macintosh. The ClustalX home-site guarantees the latest version (*see* CD and Website: http://www-igbmc.u-strasbg.fr/BioInfo/ClustalX/). Complete documentation comes with the program and is accessed through a *Help* menu.

See companion CD

WWW



```
1  Giardia_l
2  Diplomonad_sp
3  Hexamita_i
4  Spironucleus_v
5  Spathidium_sp
6  Cryptosporidiu
7  Plasmodium_k
8  Plasmodium_b
9  Plasmodium_f
10 Euplotes_a
11 Euplotes_c
12 Blastocystis_h
13 Stentor_c
14 Blepharisma_j
15 Entamoeba_h
16 Oxymonadida_sp
17 Dinenympha_e
18 Trypanosoma_c
19 Trypanosoma_b
20 Kentrophoros_s
21 Euglena_g
22 Planoprotostel
23 Dictyostelium_
24 Physarum_p
25 Cyanophora_p
26 Phytophthora_i
27 Stylonychia_m
28 Stylonychia_l
29 Paranophrys_c
   ruler     1.......10........20........30........40........50........60........70........80........90.......100.......110.......
```

File BioInfo 4U:Curricula:MSA Textbook Chapters:nogap.msf loaded.

Fig. 2. The ClustalX interface with a newly loaded dataset ready to be aligned.

Running ClustalX on Your Machine, Briefly

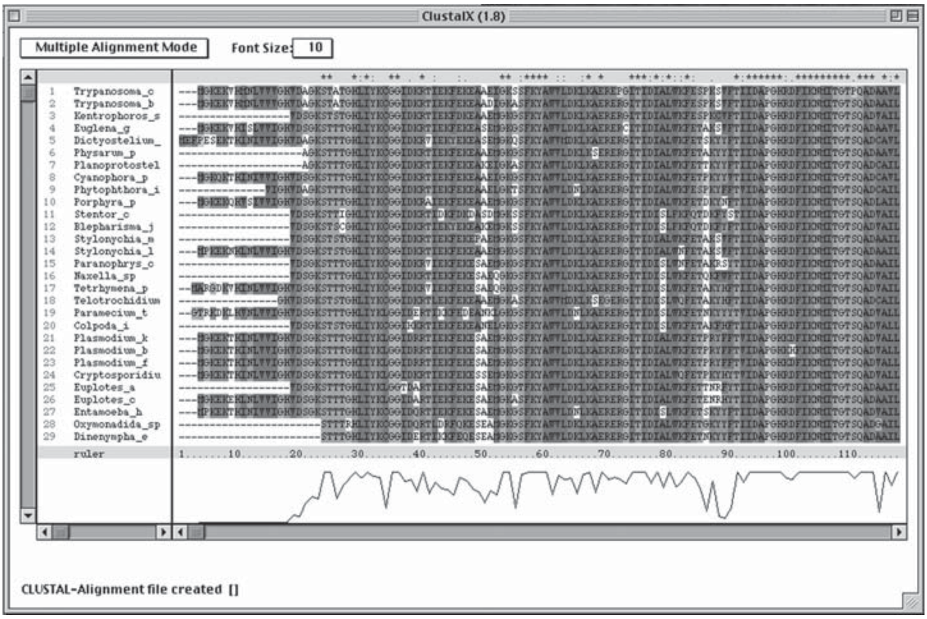
Sequences are entered into the program through the *File* menu *Load Sequences* choice. They must all be in one of the following file formats: NBRF/PIR, EMBL/SWISSPROT, Pearson's FastA, Smith's GDE, GCG's MSF, or RSF, or its own native Clustal format. The GenBank format cannot be used. Therefore, if you are saving sequences from the National Center for Biotechnology Information (NCBI), the home of GenBank, be careful to switch the default saved sequence format to be compatible with ClustalX. However, be careful of FastA format files downloaded from NCBI. For example, ClustalX will not properly load FastA format sequences, uploaded from a Mac running Netscape using Entrez at NCBI. NCBI encodes UNIX style carriage returns in the file and Mac ClustalX requires Mac style hard returns. If you run into this problem, BBEdit can be used to change the return type to Mac style globally in the file and it will work fine thereafter.

The collection of sequences used in the example contains representative EF-1α sequences from many *lower* eukaryotes. This dataset was assembled using GCG's LookUp program, a Sequence Retrieval System (SRS) derivative. These sequences can also be collected using Entrez at NCBI, either through the Web or installed as their client/server NetEntrez application, or SRS on the Web, available at all EMBL and many other biocomputing sites around the world (*see* Website: http://srs.ebi.ac.uk/). After the sequences are properly loaded into ClustalX, the window should look similar to Fig. 2.

WWW

See companion CD for color Fig. 2

Colors are based on the physical properties of the amino acids and can easily be modified by changing the default color parameter file. The plot along the bottom shows positional conservation within the dataset. Since this dataset is not yet aligned, there is little positional conservation at this point.



```
1   Trypanosoma_c
2   Trypanosoma_b
3   Kentrophoros_s
4   Euglena_g
5   Dictyostelium_
6   Physarum_p
7   Planoprotostel
8   Cyanophora_p
9   Phytophthora_i
10  Porphyra_p
11  Stentor_c
12  Blepharisma_j
13  Stylonychia_m
14  Stylonychia_l
15  Paranophrys_c
16  Naxella_sp
17  Tetrhymena_p
18  Telotrochidium
19  Paramecium_t
20  Colpoda_i
21  Plasmodium_k
22  Plasmodium_b
23  Plasmodium_f
24  Cryptosporidiu
25  Euplotes_a
26  Euplotes_c
27  Entamoeba_h
28  Oxymonadida_sp
29  Dinenympha_e
    ruler   1.......10........20........30........40........50........60........70........80........90.......100.......110.......
```

CLUSTAL-Alignment file created []

Fig. 3. An aligned dataset in ClustalX. Notice the columns of color, as well as the similarity plot along the bottom, and the similarity symbols along the top of the display.

The ClustalX interface is not an editor. You can select entire sequence entries and cut and paste them from the alignment or change their order but you can not change any characters within the alignment including gaps. The exception is you can globally remove gaps, either all of them or just common columns of gaps. Any manual editing of the alignment is not possible. This can only be done with a dedicated sequence editor, such as GCG's SeqLab editor (*see* Chapters 28 and 33).

The *Help* pages within the program are quite valuable. Documentation on the Web, including tutorials (*see* Website: http://www.may.ie/academic/biology/james/ClustalX_tutorial.html as an example) are also available. ClustalW uses an enhanced progressive, pairwise, dynamic, programming-alignment algorithm. Enhancements include the addition of an alternative fast heuristic-alignment method, differential sequence weighting, the use of a neighbor-joining guide tree rather than a UPGMA and the use of several additional protein parameters over and above the built in scoring matrices, such as different gap penalties associated with runs of hydrophilic amino acids or other structural attributes. Look through all of the program parameters and think about each before you run your alignments.

To create an alignment using default parameters **Select All Sequences** using the **Edit** menu and then **Do Complete Alignment** from the **Alignment** menu. However, take a moment to review the **Alignment Parameters** from the **Alignment** menu before starting the process. Your machine will dedicate itself to solving the alignment after pressing **ALIGN** from the *Complete Alignment* window. This dataset took under 5 min on a 300 MHz G3 Macintosh computer to compute using the default slow alignment method. The results are shown in Fig. 3.

See companion CD for color Fig. 3

Now the colors align nicely and a line appears along the top where asterisks (*) indicate absolute residue conservation, semi-colons (:) denote strong conservation, and periods (.) show weak conservation. The cutoff value between strong and weak amino acid relationships is ± 0.5 from the built-in Gonnet PAM250 scoring matrix. The plot along the bottom of peaks indicates those areas of sequence conservation and valleys show areas of sequence divergence (or misalignment). Alignment quality can also be gauged with the options **Calculate**, **Show Low-Scoring Segments**, and **Show Exceptional Residues** that highlight residues in black and gray respectively. Regions of low quality can be realigned by selecting across their columns, lowering the **Multiple Alignment Parameters**, **Gap Opening** penalty and then using the **Realign Selected Residue Range** option. This can be very effective for *cleaning* up problem areas of the alignment.

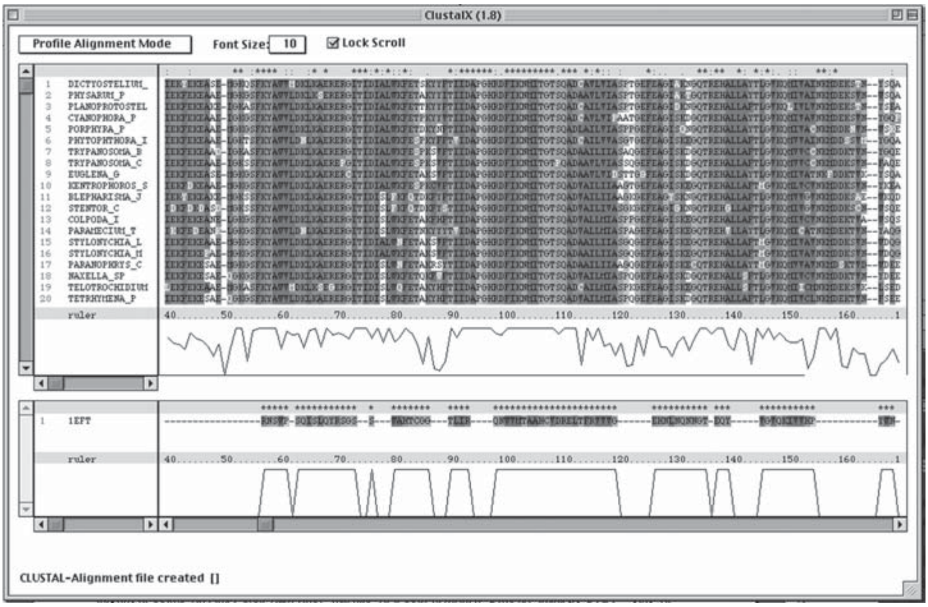
Clustal also lets you manipulate two alignments, or an existing alignment and other unaligned sequences, all at once in the *Profile Alignment Mode*. A Clustal profile is just a multiple sequence alignment, but it does provide an efficient and effective means for combining alignments or adding to an existing alignment. To run ClustalX in this manner switch from *Multiple Alignment Mode* to *Profile Alignment Mode*. This can be illustrated by adding the *T. aquaticus* EF1-Tu sequence to the example dataset. After switching modes, **Load Profile 2** to the empty lower panel, then **Align Profile 2 to Profile 1** from the **Alignment** menu. If you have changed any alignment parameters, while refining some region, e.g., you may have lowered gap penalties, you may want to reset them for this step. After pressing **ALIGN**, the new sequence will be aligned to the existing multiple alignment. The **Lock Scroll** button allows you to simultaneously scroll through both alignments with one scroll bar. *Profile Alignment Mode* is displayed in Fig. 4. To merge the pregapped 1EFT sequence with the existing alignment, you need to use the **Edit** menu to **Add Profile 2 to Profile 1** and then **Save Profile 1 As...** a new alignment file. Increasing an alignment's size in this manner is not nearly as powerful and accurate as the true profile methods. However, it is certainly an easy-to-use alternative that works quite well as long as Profile 1 and Profile 2 are sufficiently simil**ar.**

See companion CD for color Fig. 4

The World Wide Web and Multiple Sequence Alignment

Web resources for multiple alignments are not as easy to use nor as powerful as locally performing the multiple alignment, on either your own office machine or on a local dedicated sequence analysis server. Some of the difficulty comes from limits in Web interface scripting and forms capabilities, and cut-and-paste errors, but also just the unreliability of Internet connections. In spite of that warning, it is possible, and straightforward to take advantage of multiple sequence resources available on the Internet through the www. Naturally, if you are not willing to install and use any local tools for multiple alignment, www resources can be invaluable. However, problems with very large datasets make multiple sequence alignment on the Web impractical after your data has reached a certain size.

One of the most comprehensive multiple sequence alignment resource collections on the www are at the Bielefeld University Virtual School of Natural Sciences BioComputing Division (VSNS-BCD) in Germany. This is part of an extensive Web site developed, starting in 1995, for teaching bioinformatics over the Internet. Exploring all of their pages is very worthwhile as it contains incredibly informative lectures, demonstrations, and tutorials. The VSNS-BCD multiple alignment URL home page



ClustalX (1.8)

Profile Alignment Mode

Font Size: 10

☑ Lock Scroll

CLUSTAL-Alignment file created []

Fig. 4. ClustalX in *Profile Alignment Mode*. A new elongation factor sequence is merged with the existing alignment in this example.

(*see* Website http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/welcome.html) is shown in Fig. 5. Another very good multiple alignment resource is at the PBIL (Pôle Bio-Informatique Lyonnais) World Wide Web server in Lyon, France (*see* Website: http://pbil.univ-lyon1.fr/alignment.html). Developed in association with the Laboratory of Biometry and Evolutionary Biology and the Institute of Biology and Chemistry of Proteins, the PBIL multiple alignment page has some of the same URLs as the VSNS page, but the differences merit a look. Its multiple alignment section is displayed in Fig. 6.

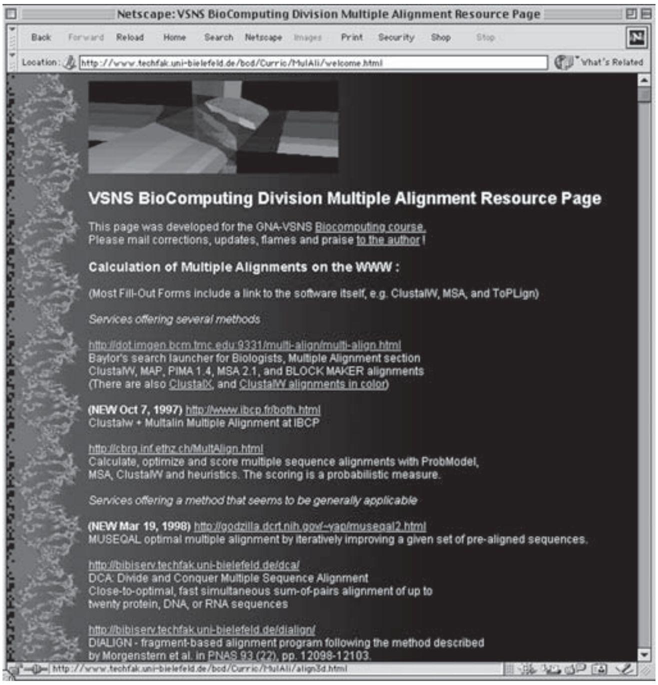
See companion CD for color Figs. 5 and 6

ClustalW on the Web

The European Molecular Biology Laboratory's European Bioinformatics Institute (EMBL EBI) in Hinxton UK provides an interface to ClustalW. All program parameters are set from the main forms window (*see* Website: http://www.ebi.ac.uk/clustalw/), and are shown in Fig. 7. To run ClustalW through a Web server in the USA go to the Baylor College of Medicine Search Launcher (*see* Website: http://searchlauncher.bcm.tmc.edu/). The Baylor Search Launcher provides a single, powerful Web portal to many different types of sequence analysis services available on the WWW. Functions are organized according to what type of analysis is offered, e.g., protein vs DNA similarity searches. The welcome page is displayed in Fig. 8.

See companion CD for color Figs. 7 and 8

Click on **Multiple sequence alignments** to get the correct form and paste your unaligned sequence set in the box. Many popular formats are accepted, though the FastA format is probably the most reliable. Select the **Most ReadSeq formats accepted** link to read the help file for all of your input choices. Program Help files are displayed, Options are set, default Parameters are listed, and Examples are shown, respectively, by



http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/welcome.html

VSNS BioComputing Division Multiple Alignment Resource Page

This page was developed for the GNA-VSNS Biocomputing course.
Please mail corrections, updates, flames and praise to the author !

Calculation of Multiple Alignments on the WWW :

(Most Fill-Out Forms include a link to the software itself, e.g. ClustalW, MSA, and ToPLign)

Services offering several methods

http://dot.imgen.bcm.tmc.edu:9331/multi-align/multi-align.html
Baylor's search launcher for Biologists, Multiple Alignment section
ClustalW, MAP, PIMA 1.4, MSA 2.1, and BLOCK MAKER alignments
(There are also ClustalX, and ClustalW alignments in color)

(NEW Oct 7, 1997) http://www.ibcp.fr/both.html
Clustalw + Multalin Multiple Alignment at IBCP

http://cbrg.inf.ethz.ch/MultAlign.html
Calculate, optimize and score multiple sequence alignments with ProbModel,
MSA, ClustalW and heuristics. The scoring is a probabilistic measure.

Services offering a method that seems to be generally applicable

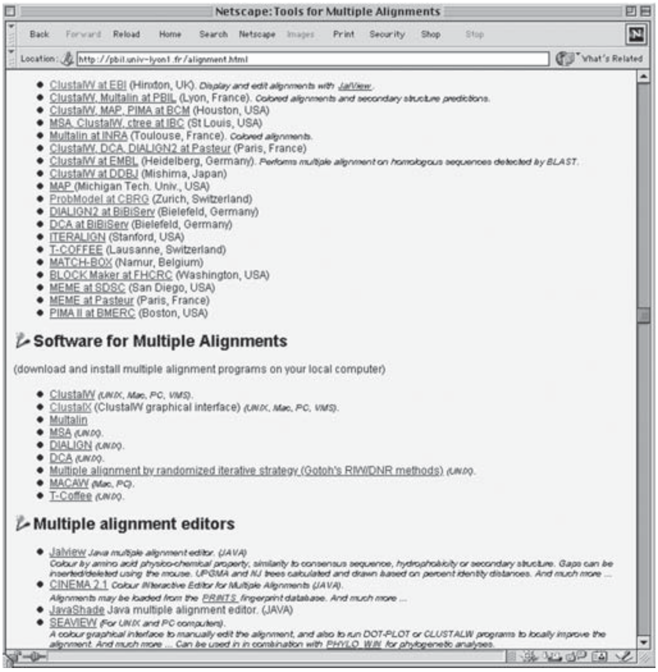
(NEW Mar 19, 1998) http://godzilla.dcrt.nih.gov/~yap/museqal2.html
MUSEQAL optimal multiple alignment by iteratively improving a given set of pre-aligned sequences.

http://bibiserv.techfak.uni-bielefeld.de/dca/
DCA: Divide and Conquer Multiple Sequence Alignment
Close-to-optimal, fast simultaneous sum-of-pairs alignment of up to
twenty protein, DNA, or RNA sequences

http://bibiserv.techfak.uni-bielefeld.de/dialign/
DIALIGN - fragment-based alignment program following the method described
by Morgenstern et al. in PNAS 93 (22), pp. 12098-12103.

http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/align3d.html

Fig. 5. The VSNS-BCD multiple sequence alignment home page (*see* Website: http://www.techfak.uni-bielefeld.de/bcd/Curric/MulAli/welcome.html).



http://pbil.univ-lyon1.fr/alignment.html

- ClustalW at EBI (Hinxton, UK). *Display and edit alignments with JalView.*
- ClustalW, Multalin at PBIL (Lyon, France). *Colored alignments and secondary structure predictions.*
- ClustalW, MAP, PIMA at BCM (Houston, USA)
- MSA, ClustalW, ctree at IBC (St Louis, USA)
- Multalin at INRA (Toulouse, France). *Colored alignments.*
- ClustalW, DCA, DIALIGN2 at Pasteur (Paris, France)
- ClustalW at EMBL (Heidelberg, Germany). *Performs multiple alignment on homologous sequences detected by BLAST.*
- ClustalW at DDBJ (Mishima, Japan)
- MAP (Michigan Tech. Univ., USA)
- ProbModel at CBRG (Zurich, Switzerland)
- DIALIGN2 at BiBiServ (Bielefeld, Germany)
- DCA at BiBiServ (Bielefeld, Germany)
- ITERALIGN (Stanford, USA)
- T-COFFEE (Lausanne, Switzerland)
- MATCH-BOX (Namur, Belgium)
- BLOCK Maker at FHCRC (Washington, USA)
- MEME at SDSC (San Diego, USA)
- MEME at Pasteur (Paris, France)
- PIMA II at BMERC (Boston, USA)

Software for Multiple Alignments

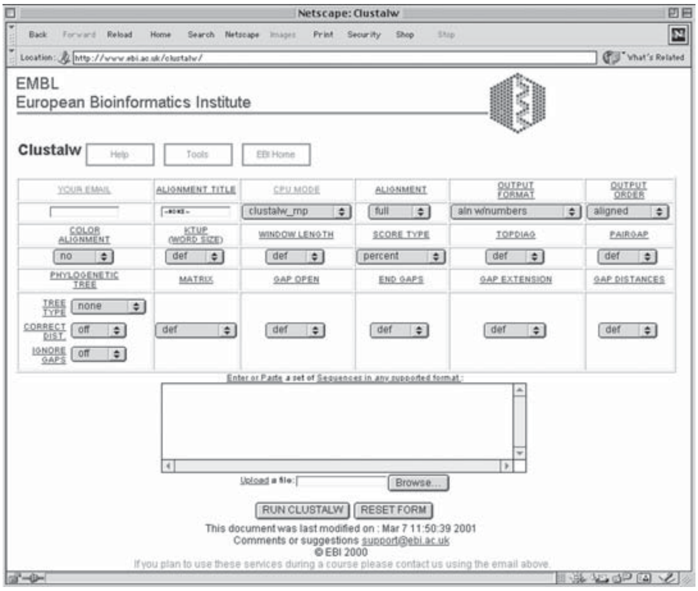
(download and install multiple alignment programs on your local computer)

- ClustalW *(UNIX, Mac, PC, VMS).*
- ClustalX (ClustalW graphical interface) *(UNIX, Mac, PC, VMS).*
- Multalin
- MSA *(UNIX).*
- DIALIGN *(UNIX).*
- DCA *(UNIX).*
- Multiple alignment by randomized iterative strategy (Gotoh's RIW/DNR methods) *(UNIX).*
- MACAW *(Mac, PC).*
- T-Coffee *(UNIX).*

Multiple alignment editors

- Jalview *Java multiple alignment editor. (JAVA)*
  Colour by amino acid physico-chemical property, similarity to consensus sequence, hydrophobicity or secondary structure. Gaps can be inserted/deleted using the mouse. UPGMA and NJ trees calculated and drawn based on percent identity distances. And much more ...
- CINEMA 2.1 *Colour INteractive Editor for Multiple Alignments (JAVA).*
  Alignments may be loaded from the PRINTS fingerprint database. And much more ...
- JavaShade Java multiple alignment editor. (JAVA)
- SEAVIEW *(For UNIX and PC computers).*
  A colour graphical interface to manually edit the alignment, and also to run DOT-PLOT or CLUSTALW programs to locally improve the alignment. And much more ... Can be used in in combination with PHYLO_WIN for phylogenetic analyses.

Fig. 6. The Pôle Bio-Informatique Lyonnais (PBIL) World Wide Web server's alignment page (*see* Website: http://pbil.univ-lyon1.fr/alignment.html).



EMBL
European Bioinformatics Institute

Clustalw

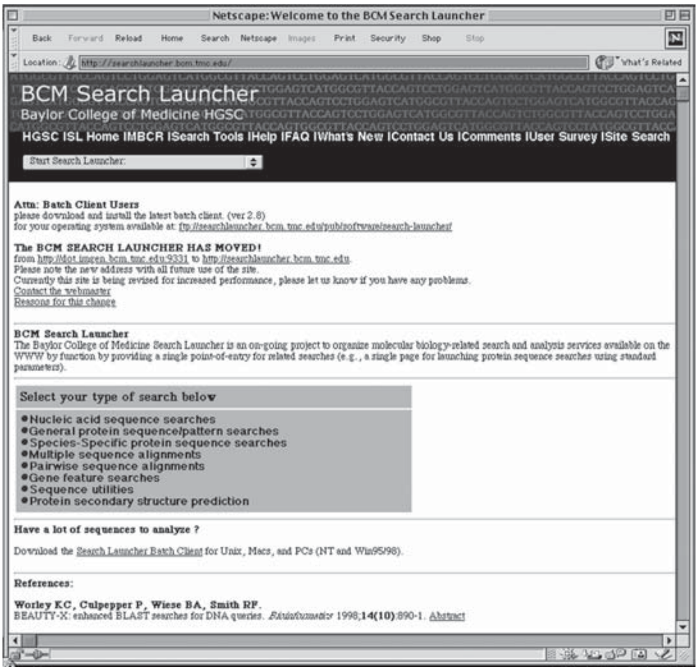
YOUR EMAIL	ALIGNMENT TITLE	CPU MODE	ALIGNMENT	OUTPUT FORMAT	OUTPUT ORDER
	-NONE-	clustalw_mp	full	aln w/numbers	aligned
COLOR ALIGNMENT	KTUP (WORD SIZE)	WINDOW LENGTH	SCORE TYPE	TOPDIAG	PAIRGAP
no	def	def	percent	def	def
PHYLOGENETIC TREE	MATRIX	GAP OPEN	END GAPS	GAP EXTENSION	GAP DISTANCES
TREE TYPE none CORRECT DIST. off IGNORE GAPS off	def	def	def	def	def

Enter or Paste a set of Sequences in any supported format:

Upload a file:

This document was last modified on : Mar 7 11:50:39 2001
Comments or suggestions support@ebi.ac.uk
© EBI 2000
If you plan to use these services during a course please contact us using the email above.

Fig. 7. The European Molecular Biology Laboratory's European Bioinformatics Institute's World Wide Web interface to ClustalW. Program parameters are all set right from the main form window (*see* Website: http://www.ebi.ac.uk/clustalw/).


Netscape: Welcome to the BCM Search Launcher
Back
Forward
Reload
Home
Search
Netscape
Images
Print
Security
Shop
Stop
Location: http://searchlauncher.bcm.tmc.edu/
What's Related
HGSC ISL Home IMBCR ISearch Tools IHelp IFAQ IWhat's New IContact Us IComments IUser Survey ISite Search
Start Search Launcher:

BCM Search Launcher
Baylor College of Medicine HGSC

Attn: Batch Client Users
please download and install the latest batch client. (ver 2.8)
for your operating system available at: ftp://searchlauncher.bcm.tmc.edu/pub/software/search-launcher/

The BCM SEARCH LAUNCHER HAS MOVED!
from http://dot.imgen.bcm.tmc.edu:9331 to http://searchlauncher.bcm.tmc.edu.
Please note the new address with all future use of the site.
Currently this site is being revised for increased performance, please let us know if you have any problems.
Contact the webmaster
Reasons for this change

BCM Search Launcher
The Baylor College of Medicine Search Launcher is an on-going project to organize molecular biology-related search and analysis services available on the WWW by function by providing a single point-of-entry for related searches (e.g., a single page for launching protein sequence searches using standard parameters).

Select your type of search below

- Nucleic acid sequence searches
- General protein sequence/pattern searches
- Species-Specific protein sequence searches
- Multiple sequence alignments
- Pairwise sequence alignments
- Gene feature searches
- Sequence utilities
- Protein secondary structure prediction

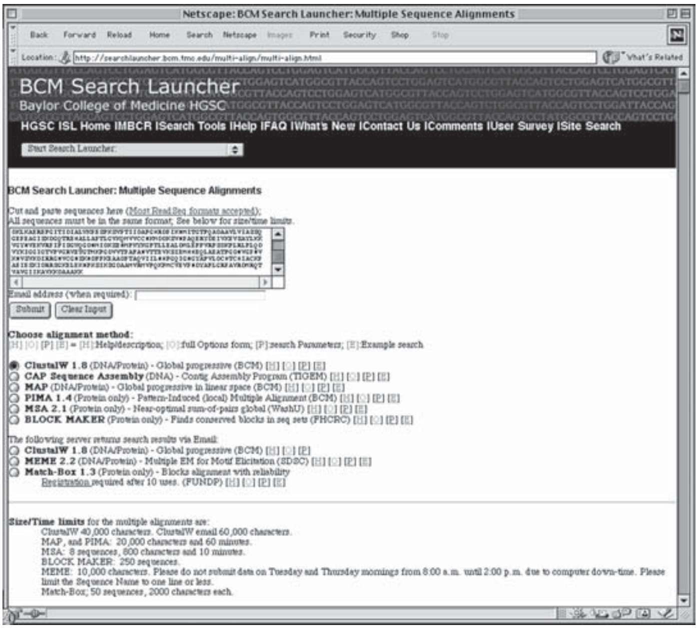
Have a lot of sequences to analyze ?

Download the Search Launcher Batch Client for Unix, Macs, and PCs (NT and Win95/98).

References:

Worley KC, Culpepper P, Wiese BA, Smith RF.
BEAUTY-X: enhanced BLAST searches for DNA queries. *Bioinformatics* 1998;**14(10)**:890-1. Abstract

Fig. 8. The Baylor College of Medicine's BCM Search Launcher welcome page (*see* Website: http://searchlauncher.bcm.tmc.edu/). The Baylor Search Launcher organizes commonly used search functions according to what type of analysis is offered, e.g. protein versus DNA similarity searches.


Netscape: BCM Search Launcher: Multiple Sequence Alignments
Back
Forward
Reload
Home
Search
Netscape
Images
Print
Security
Shop
Stop
Location:
http://searchlauncher.bcm.tmc.edu/multi-align/multi-align.html
What's Related
BCM Search Launcher
Baylor College of Medicine HGSC
HGSC ISL Home IMBCR ISearch Tools IHelp IFAQ IWhat's New IContact Us IComments IUser Survey ISite Search
Start Search Launcher:

BCM Search Launcher: Multiple Sequence Alignments

Cut and paste sequences here (Most ReadSeq formats accepted):
All sequences must be in the same format; See below for size/time limits.

Email address (when required):

Submit | Clear Input

Choose alignment method:
[H] [O] [P] [E] = [H]:Help/description; [O]:full Options form; [P]:search Parameters; [E]:Example search

● **ClustalW 1.8** (DNA/Protein) - Global progressive (BCM) [H] [O] [P] [E]
○ **CAP Sequence Assembly** (DNA) - Contig Assembly Program (TIGEM) [H] [O] [P] [E]
○ **MAP** (DNA/Protein) - Global progressive in linear space (BCM) [H] [O] [P] [E]
○ **PIMA 1.4** (Protein only) - Pattern-Induced (local) Multiple Alignment (BCM) [H] [O] [P] [E]
○ **MSA 2.1** (Protein only) - Near-optimal sum-of-pairs global (WashU) [H] [O] [P] [E]
○ **BLOCK MAKER** (Protein only) - Finds conserved blocks in seq sets (FHCRC) [H] [O] [P] [E]

The following server returns search results via Email:
○ **ClustalW 1.8** (DNA/Protein) - Global progressive (BCM) [H] [O] [P]
○ **MEME 2.2** (DNA/Protein) - Multiple EM for Motif Elicitation (SDSC) [H] [O] [P] [E]
○ **Match-Box 1.3** (Protein only) - Blocks alignment with reliability
Registration required after 10 uses. (FUNDP) [H] [O] [P] [E]

Size/Time limits for the multiple alignments are:
ClustalW 40,000 characters. ClustalW email 60,000 characters.
MAP, and PIMA: 20,000 characters and 60 minutes.
MSA: 8 sequences, 800 characters and 10 minutes.
BLOCK MAKER: 250 sequences.
MEME: 10,000 characters. Please do not submit data on Tuesday and Thursday mornings from 8:00 a.m. until 2:00 p.m. due to computer down-time. Please limit the Sequence Name to one line or less.
Match-Box; 50 sequences, 2000 characters each.

Fig. 9. The Baylor College of Medicine's BCM Search Launcher's multiple sequence alignment input form.

See companion CD for color Fig. 9

clicking the *HOPE* buttons next to every service: [**H**] [**O**] [**P**] [**E**]. The default multiple alignment program choice is ClustalW. The screen will look similar to that shown in Fig. 9 after you've pasted your dataset into the form. Press the **Submit** button when ready. The Search Launcher screen will go away and be replaced by the following text:

```
          ClustalW Multiple Sequence Alignment Results
               Courtesy of the BCM Search Launcher
Your job has been accepted. Please DO NOT submit it again. Thanks.
         Click here to check your job cw18-3028 status.
```

The sample dataset was aligned in less than 5 min. It is returned in an easy-to-read interleaved format as well as a standard FastA format section that you can cut-and-paste into subsequent applications. The alignment is very similar to that derived on the desktop. There are some interesting differences though. In particular the Baylor ClustalW alignment is considerably shorter, 471 residues and gaps long, vs 498. One significant difference, ClustalX uses the Gonnet scoring matrix series by default whereas ClustalW through the Baylor Search Launcher uses the BLOSUM matrix series by default.

Web Alternatives to ClustalW

MSA and PIMA are both very specialized programs. Generally, they would not be used for most standard datasets and in fact, both have quite limited input capacities. So why would anyone want to use either? It is largely a matter of local vs global similarity and dissatisfaction with standard algorithms. PIMA is very good at finding local *patches* of similarity between sequences and aligning those patches independent of high background dissimilarity. MSA globally aligns the full length of sequences but does so *all-at-the-same-time*, it does not use a pairwise process, it explores an n-dimensional trace-back matrix. So, PIMA is valuable for aligning domains of proteins where full-length alignments just will not work, and MSA is good for aligning the full lengths of proteins in situations where pairwise, progressive techniques leave too much ambiguity. However, both programs are limited to small datasets. The PIMA limit at the Baylor Search Launcher is 20,000 total characters and 60 min of computation time whereas the MSA limit is 8 sequences, 800 characters, and 10 min of computation time.

Multiple Sequence Alignment and Structure Prediction

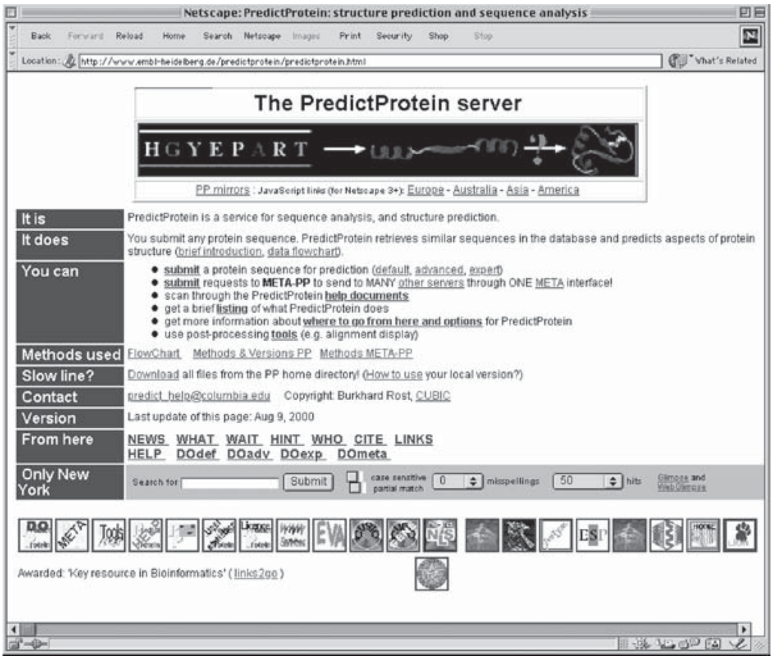
Structural inference is fraught with difficulties. However, it becomes possible using comparative multiple sequence approaches. One of the best predictors of secondary structure is on the World Wide Web (*see* Website: http://www.embl-heidelberg.de/predictprotein/predictprotein.html). This uses multiple sequence alignment profile techniques along with neural net technology. PredictProtein is a service offered by the Protein Design Group at the European Molecular Biology Laboratory, Heidelberg, Germany. A multiple sequence alignment is created with the MaxHom weighted dynamic programming method and a secondary structure prediction is produced by the profile network method (PHD). PHD is rated at an expected 70.2% average accuracy for the three states helix, strand, and loop. This Web page provides default, advanced, and expert submission forms. One powerful advanced and expert option is the ability to submit your own multiple alignments. Their automated search and alignment procedure is very good, but if you have been working for months on a multiple alignment, and you know it is the best it can be, you may want to force PredictProtein to use that information, rather than it's own automated alignment. The welcome page shown in Fig. 10 presents a wealth of informational links.

See companion CD for color Fig. 10

Three-dimensional modeling without crystal coordinates is even possible. This is *homology modeling*. It will often lead to remarkably accurate representations if the similarity is great enough between your protein and one with an experimentally solved structure. Automated homology modeling is available through the Web as GlaxoSmithKline's SWISS-MODEL at Amos Bairoch's ExPASy server in Switzerland (*see* Website: http://www.expasy.ch/swissmod/SWISS-MODEL.html). As with PredictProtein, using the *First Approval Mode* you can submit an individual sequence and the server will perform a database search. In this case it will search against all of the sequences from the three-dimensional Protein Data Bank. It will then create a multiple alignment of the significant hits, to provide a structural inference. Alternatively, using the *Optimise (project) mode* you can submit your own customized and carefully scrutinized multiple sequence alignment. Naturally, your template sequences must have solved structures. Swiss-PdbViewer must be used to format and submit your data. Swiss-PdbViewer is an interactive molecular structure viewer and editor developed at GlaxoSmithKline. Swiss-PdbViewer allows superpositioning of both structures and their corresponding sequences, that you install on your own computer. It has versions for many of the major operating systems. An extensive menu and help system is provided by the SWISS-MODEL home page as shown in Fig. 11.

See companion CD for color Fig. 11

Results are returned via e-mail in one of three modes, Swiss-PdbViewer mode, normal mode, or short mode. Normal mode and short mode both return PDB format coordinates for the model, normal with a complete log file of all the server actions,


Netscape: PredictProtein: structure prediction and sequence analysis
Back Forward Reload Home Search Netscape Images Print Security Shop Stop
Location: http://www.embl-heidelberg.de/predictprotein/predictprotein.html
What's Related
HGYEPART

The PredictProtein server

PP mirrors : JavaScript links (for Netscape 3+): Europe - Australia - Asia - America

It is: PredictProtein is a service for sequence analysis, and structure prediction.

It does: You submit any protein sequence. PredictProtein retrieves similar sequences in the database and predicts aspects of protein structure (brief introduction, data flowchart).

You can:

- submit a protein sequence for prediction (default, advanced, expert)
- submit requests to **META-PP** to send to MANY other servers through ONE META interface!
- scan through the PredictProtein help documents
- get a brief listing of what PredictProtein does
- get more information about where to go from here and options for PredictProtein
- use post-processing tools (e.g. alignment display)

Methods used: FlowChart Methods & Versions PP Methods META-PP

Slow line?: Download all files from the PP home directory! (How to use your local version?)

Contact: predict_help@columbia.edu Copyright: Burkhard Rost, CUBIC

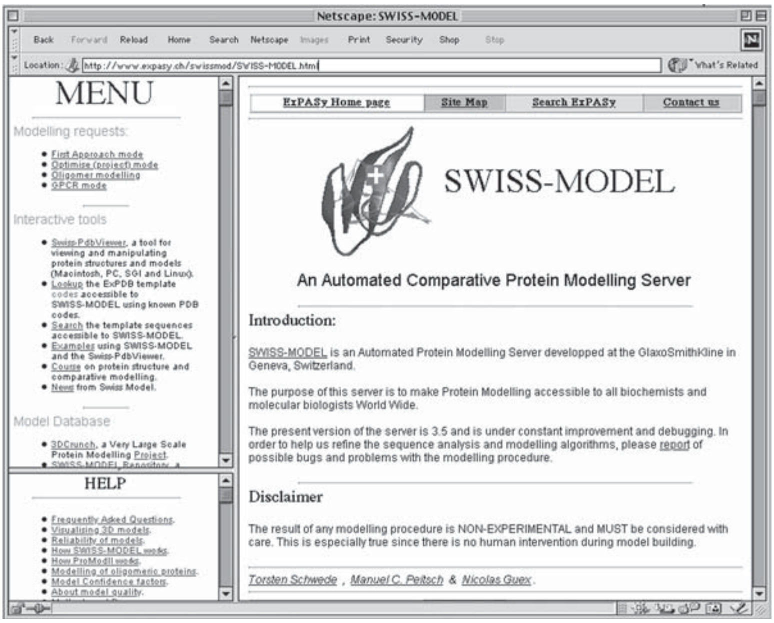
Version: Last update of this page: Aug 9, 2000

From here: NEWS WHAT WAIT HINT WHO CITE LINKS HELP DOdef DOadv DOexp DOmeta

Only New York: Search for □ Submit □ case sensitive □ partial match 0 misspellings 50 hits Glimpse and WebGlimpse

Awarded: 'Key resource in Bioinformatics' (links2go)

Fig. 10. The PredictProtein secondary structure prediction server home page in Heidelberg, Germany (*see* Website: http://www.embl-heidelberg.de/predictprotein/predictprotein.html).


Netscape: SWISS-MODEL
Back Forward Reload Home Search Netscape Images Print Security Shop Stop
Location: http://www.expasy.ch/swissmod/SWISS-MODEL.html
What's Related

MENU

Modelling requests:

- First Approach mode
- Optimise (project) mode
- Oligomer modelling
- GPCR mode

Interactive tools

- Swiss-PdbViewer, a tool for viewing and manipulating protein structures and models (Macintosh, PC, SGI and Linux).
- Lookup the ExPDB template codes accessible to SWISS-MODEL using known PDB codes.
- Search the template sequences accessible to SWISS-MODEL.
- Examples using SWISS-MODEL and the Swiss-PdbViewer.
- Course on protein structure and comparative modelling.
- News from Swiss Model.

Model Database

- 3DCrunch, a Very Large Scale Protein Modelling Project.
- SWISS-MODEL Repository, a

HELP

- Frequently Asked Questions.
- Visualising 3D models.
- Reliability of models.
- How SWISS-MODEL works.
- How ProModII works.
- Modelling of oligomeric proteins.
- Model Confidence factors.
- About model quality.

ExPASy Home page | Site Map | Search ExPASy | Contact us

SWISS-MODEL

An Automated Comparative Protein Modelling Server

Introduction:

SWISS-MODEL is an Automated Protein Modelling Server developped at the GlaxoSmithKline in Geneva, Switzerland.

The purpose of this server is to make Protein Modelling accessible to all biochemists and molecular biologists World Wide.

The present version of the server is 3.5 and is under constant improvement and debugging. In order to help us refine the sequence analysis and modelling algorithms, please report of possible bugs and problems with the modelling procedure.

Disclaimer

The result of any modelling procedure is NON-EXPERIMENTAL and MUST be considered with care. This is especially true since there is no human intervention during model building.

Torsten Schwede , *Manuel C. Peitsch* & *Nicolas Guex* .

Fig. 11. GlaxoSmithKline's SWISS-MODEL automated homology modeling home page at Amos Bairoch's ExPASy server in Switzerland (*see* Website: http://www.expasy.ch/swissmod/SWISS-MODEL.html).

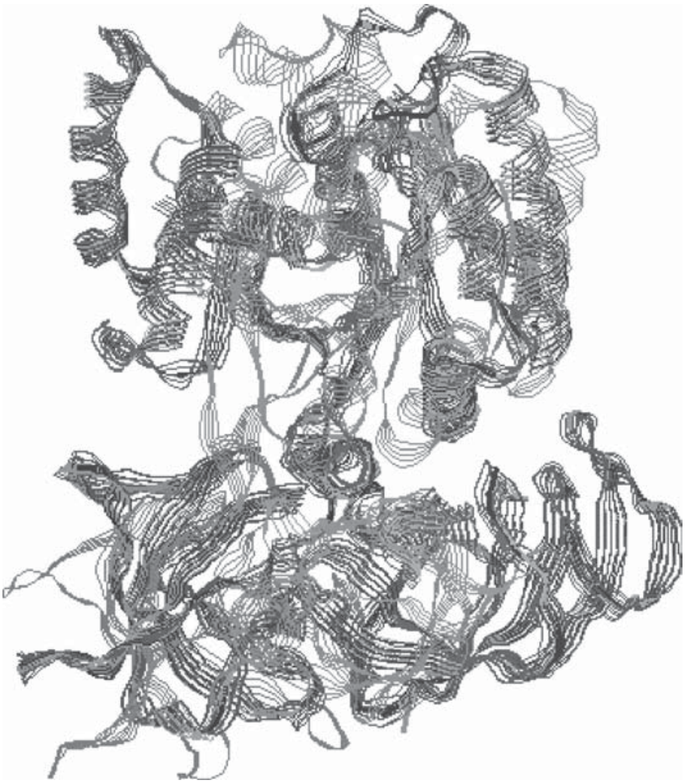


Fig. 12. A RasMac *Strands* graphic of the *Giardia* EF1α structural model superimposed over eight other chains (*see* Website: http://www.umass.edu/microbio/rasmol/).

See companion CD for color Fig. 12

WWW

short without. Swiss-PdbViewer mode returns a project file containing PDB formatted coordinates for the model and all templates superimposed, formatted for Swiss-PdbViewer, and a complete log file. The results from submitting the *Giardia lamblia* elongation factor 1α sequence to SWISS_MODEL in *First Approach mode* were e-mailed back in less than 5 min. Figure 12 displays a RasMac (*see* Website: http://www.umass.edu/microbio/rasmol/) *Strands* graphic of the *Giardia* EF1α structural model superimposed over eight other chains.

Glossary and Abbreviations

Affine An affine function is a linear function, described by the algebraic formula: y = mx + b.

Anchor An anchor in the context of a multiple sequence alignment is a region across all the sequences of the alignment that is very highly conserved and thus can help guide alignment by constraint of that region.

Crown Group In the context of the universal tree of life, the Crown group refers to the ‘explosion’ of diversity that occurred relatively late in the history of life on earth and includes all animals, plants, true fungi, algae, and many protists.

Gap Penalties The creation or opening penalty is how many points a dynamic programming algorithm is penalized for imposing a gap in an alignment. The extension or lengthening penalty describes how many additional points the algorithm is penalized for each additional gap added to the first one, after that first gap, the gap creation penalty, is introduced.

Global Alignment As opposed to local alignment, which is the alignment of only the best regions within sequences, global alignment is the alignment of the full length of your sequence set and generally applies to the multiple sequence alignment problem. However, it should be realized that global alignment can be restricted to subsequences within sequences, the distinction being that local alignment 'picks' the best regions for you, whereas global alignment uses the full length of whatever is specified.

Homology Homology, as opposed to sequence identity and similarity, can have no level. A sequence and, in fact, a position within an alignment, is either demonstrably related via evolution to another or it is not. Statistically significant similarity can argue for homology; however, a lack of statistically significant similarity can not be used to argue against homology.

Homology Modeling The secondary and often tertiary, three-dimensional structure of proteins can be inferred by alignment with proteins whose structure has been experimentally determined. Obviously the more similar the sequences are, the more successful the model will be.

Indel An indel is a gap introduced into a sequence alignment necessary to reconcile differing lengths and evolutionary histories. It is impossible to ascertain whether an insertion or a deletion event created the discrepancy, hence the term indel.

Lower or Primitive Eukaryotes Both terms are misnomers as they imply evolutionary 'progress' and these organisms are no less successful than the 'higher' eukaryotes of the Crown group, but the terms persist and are descriptive of the more basal placement of this assemblage of protists in the universal tree of life.

Matrices A match matrix is the first step in solving the dynamic programming algorithm. Its cells contain the value each position receives for matching (aligning) respective X and Y axis characters. A score matrix can have two meanings. In the context of dynamic programming it is the matrix in which cell values have received initial match values adjusted by gap penalties and trace back paths. The alternative meaning describes the values that amino acid residues or bases receive for aligning with one another, e.g. the PAM and BLOSUM matrix series. The trace-back path matrix delineates the alignments discovered by the dynamic programming algorithm; it illustrates the path through the matrix.

Orthology One of the two major classes of sequence homology exist. Orthology describes homologous sequences present in different organisms as a result of speciation processes. Major confusion can result from mixing paralogues and orthologues in the same analysis.

Paralogy One of the two major classes of sequence homology exist. Paralogy describes homologous sequences within the same organism as a result of gene duplication. As stated above, major confusion can result from mixing paralogues and orthologues in the same analysis.

Universal Tree of Life A phylogenetic tree, i.e. a graph that illustrates the relations of organisms through evolutionary time, that attempts to establish the placement of all earth's extant cellular life.

Suggested Readings

Dynamic Programming

Needleman, S. B. and Wunsch, C. D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48, 443–453.

Smith, T. F. and Waterman, M. S. (1981) Comparison of bio-sequences, Adv. Appl. Math. 2, 482–489.

Scoring Matrices

Henikoff, S. and Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. USA 89, 10,915–10,919.

Schwartz, R. M. and Dayhoff, M. O. (1979) Matrices for detecting distant relationships, in: Atlas of Protein Sequences and Structure, vol. 5, (Dayhoff, M. O., ed.), National Biomedical Research Foundation, Washington DC, pp. 353–358.

Multiple Sequence Dynamic Programming

Feng, D. F. and Doolittle, R. F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees, J. Mol. Evol. 25, 351–360

Genetics Computer Group (GCG), a part of Accelrys Inc., a subsidiary of Pharmacopeia Inc. (©1982-2002) Program Manual for the Wisconsin Package, Version 10.3. *(*http://www.accelrys.com/products/gcg-wisconsin-package).

Gupta, S. K., Kececioglu, J. D., and Schaffer, A. A. (1995) Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment, J. Comp. Biol. 2, 459–472.

Higgins, D. G., Bleasby, A. J., and Fuchs, R. (1992) CLUSTALV: improved software for multiple sequence alignment, Comp. Appl. Biol. Sci. 8, 189–191.

Smith, R. F. and Smith, T. F. (1992) Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for comparative protein modeling, Protein Eng. 5, 35–41.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994) CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice, Nucleic Acids Res. 22, 4673–4680.

Thompson, J. D., Gibson, T. J., Plewniak, F., Jeanmougin, F., and Higgins, D. G. (1997) The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools, Nucleic Acids Res. 24, 4876–4882.

Applicability

Alignment Profiles

Eddy, S. R. (1996) Hidden Markov models, Curr. Opin. Struct. Biol. 6, 361–365.

Eddy, S. R. (1998) Profile hidden Markov models, Bioinformatics 14, 755–763

Gribskov, M., Luethy, R., and Eisenberg, D. (1989) Profile analysis, in: Methods in Enzymology, vol. 183, Academic Press, San Diego, CA, pp. 146–159.

Gribskov M., McLachlan M., and Eisenberg, D. (1987) Profile analysis: detection of distantly related proteins, Proc. Natl. Acad. Sci. USA 84, 4355–4358.

Complications

File Formats

Gilbert, D. G. (1993 [C release] and 1999 [Java release]) ReadSeq, public domain software, Bioinformatics Group, Biology Department, Indiana University,

Bloomington, IN.
(*see* Website: http://iubio.bio.indiana.edu/soft/molbio/readseq/)

The Protein System

Phylogenetic Relaionships

The *E. coli* Database Collection (ECDC) The K12 chromosome, Justus-Liebig-Universitaet, Giessen, Germany.
(*see* Website: http://www.uni-giessen.de/ngx1052/ecdc.htm)

Hasegawa, M., Hashimoto, T., Adachi, J., Iwabe, N., and Miyata, T. (1993) Early branchings in the evolution of Eukaryotes: ancient divergence of Entamoeba that lacks mitochondria revealed by protein sequence data, J. Mol. Evol. 36, 380–388.

Iwabe, N., Kuma, E.-I., Hasegawa, M., Osawa, S., and Miyata, T. (1989) Evolutionary relationship of archaebacteria, eubacteria, and eukaryotes inferred from phylogenetic trees of duplicated genes, Proc. Natl. Acad. Sci. USA 86, 9355–9359.

Madsen, H. O. Poulsen, K., Dahl, O., Clark, B. F., and Hjorth, J. P. (1990) Retropseudogenes constitute the major part of the human elongation factor 1 alpha gene family, Nucleic Acids Res. 18, 1513–1516.

Rivera, M. C. and Lake, J. A. (1992) Evidence that eukaryotes and eocyte prokaryotes are immediate relatives, Science 257, 74–76.

What is Availble

Running ClustalX on Your Machine, Briefly

Etzold, T. and Argos, P. (1993) SRS–an indexing and retrieval tool for flat file data libraries, Comp. Appl. Biosci. 9, 49–57.

Gonnet, G. H., Cohen, M. A., and Benner, S. A. (1992) Exhaustive matching of the entire protein sequence database, Science 256, 1443–1145.

ClustalW on the Web

Smith, R. F., Wiese, B. A., Wojzynski, M. K., Davison, D. B., and Worley, K. C. (1996) BCM Search Launcher–an integrated interface to molecular biology data base search and analysis services available on the World Wide Web, Genome Res. 6, 454–62.

Multiple Sequence Alignment and Structure Prediction

Alignment Secondary Structure

Guex, N., Diemand, A., and Peitsch, M. C. (1999) Protein modeling for all, Trends Biochem. Sci. 24, 364–367.

Guex, N. and Peitsch, M. C. (1997) SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling, Electrophoresis 18, 2714–2723.

Rost, B. and Sander, C. (1993) Prediction of protein secondary structure at better than 70% accuracy, J. Mol. Biol. 232, 584–599.

Rost, B. and Sander, C. (1994) Combining evolutionary information and neural networks to predict protein secondary structure, Proteins 19, 55–77.

Sander, C. and Schneider, R. (1991) Database of homology-derived structures and the structural meaning of sequence alignment, Proteins 9, 56–68.

Sayle, R. A. and Milner-White, E. J. (1995) RasMol: biomolecular graphics for all, Trends Biochem. Sci. 20, 374–376.

32 3D Molecular Visualization with Protein Explorer

Eric Martz

Introduction

Visualization of macromolecular structure in three dimensions is becoming ever more important for understanding protein structure-function relationships, functional consequences of mutations, mechanisms of ligand binding, and drug design. Free molecular visualization software of excellent quality is now available. Protein Explorer is featured because it is easiest to use, yet quite powerful. It is compared with Cn3D, WebLab Viewer Lite and DeepView (SwissPDB-Viewer). In brief, among these four packages, Protein Explorer is best for visualization, WebLab is best for publication-quality printed graphics, and DeepView is the only freeware package capable of modeling operations such as mutation, homology modeling, and structural alignments. Other packages that have strong features, but are not covered here, include DINO, MAGE, PyMOL, and VMD. Chime and RasMol are discussed in the next subheading.

Protein Explorer is freeware, for Windows or Macintosh, designed to make 3D macromolecular visualization as easy as possible. Its basic features are designed for students and educators, but it also has powerful, yet easy to use features appreciated by protein structure specialists and crystallographers. Although the majority of uses for Protein Explorer involve protein structure visualization, it is also well suited to visualization of other macromolecules, such as RNA, DNA, and polysaccharides, as well as complexes, such as those between transcriptional regulatory proteins and DNA.

Protein Explorer can operate directly from the web (*see* Website: http://www.proteinexplorer.org), or can be downloaded for off-line use. Either Netscape Communicator or Internet Explorer is required along with a plug-in called MDL Chime. If you have not downloaded and installed Chime, when you attempt to use Protein Explorer, it will direct you where to do so. Table 1 provides a list of associated websites.

Protein Explorer vs Chime and RasMol

Because of its greater ease of use and power, Protein Explorer has superceded Chime and RasMol for interactive exploration of macromolecular structure in most situations (*see* Website: http://www.umass.edu/microbio/chime/explorer/pe_v_ras.htm). Protein Explorer is, in part, a user interface to MDL Chime that would not be possible without the visualization power and chemical intelligence built into Chime. Protein

Table 1
Major Websites

Website name	*Location*
Protein Explorer[a]	http://www.proteinexplorer.org
World Index of Molecular Visualization Resources (including tutorials using Chime, dedicated to specific molecules)	http://www.molvisindex.org
PDB Lite (search interface for novices)	http://www.pdblite.org
Protein Data Bank (source of all published macromolecular structure data)	http://www.pdb.org
Nature and Limitations of 3-D Structural Data	http://www.rcsb.org/pdb/ experimental_methods.html
Cn3D (*see* category "Free Software")	http://www.molvisindex.org
DeepView (SwissPDB-Viewer)	http://www.molvisindex.org
WebLab Viewer Lite	http://www.molvisindex.org

[a] Consult Protein Explorer's Help, Index and Glossary to find how to access specific features of Protein Explorer. It is linked to Protein Explorer's FrontDoor under About PE.

Explorer's purpose is to make the power in Chime accessible and to extend it by adding higher level capabilities (*see* Website: http://www.umass.edu/microbio/chime/explorer/why_pe.htm).

Chime is a Netscape plugin that renders images of molecules. Chime's name is derived from *Chemical* and *MIME*, where MIME denotes the method used to categorize information sent through the Internet (Multi-part Internet Mail Extension). Chime is freeware created at MDL Information Systems (*see* Website: http://www.mdlchime.com), and is built upon the rendering and command language source code of RasMol (*see* Website: http://www.umass.edu/microbio/rasmol), a brilliant and deservedly popular, open-source, stand-alone molecular visualization program generously put in the public domain by its author, Roger Sayle. In addition to its browser plugin functionality, Chime's visualization capabilities go considerably beyond those of RasMol, to include surfaces and animation support.

Those willing to acquire some programming skills can create user-friendly web pages in Chime. Examples of excellent, molecule-specific tutorials in Chime, as well as methods for building these tools are indexed (*see* Website: http://www.molvisindex.org).

Sources of Macromolecular Structure Data: The Protein Data Bank (PDB)

PDB Files and Codes

In order to display a macromolecule, Protein Explorer like the other programs, requires an *atomic coordinate data file*, that specifies the positions of each atom in space with Cartesian coordinates. There are many formats for such data files, but the most common is the Protein Data Bank (PDB) format. These files are often referred to simply as *PDB files* rather than the generic name, atomic coordinate files. A PDB file is a plain text file that can be edited with a text editor or word processor. An introduction to the PDB format is available (*see* Website: http://www.umass.edu/microbio/rasmol/pdb.htm), along with the official PDB format specification (*see* Website: http://www.pdb.org and click on **File Formats**).

Each entry deposited at the PDB is assigned a unique four-character identification code. The first character of the code is always a numeral; the last three characters can be either letters or numerals. Examples of PDB identification codes are given in Table 2. These codes can be entered in the slot on the FrontDoor page of Protein Explorer to see the corresponding molecules.

The PDB file format is antiquated and inadequate in many respects. Therefore, the International Union of Crystallography has adopted a new format called *macromolecular Crystallographic Information File Format* or *mmCIF*. However, many software packages depend on the old PDB format. To accommodate the transition, the Protein Data Bank will continue to provide all entries in PDB format for the foreseeable future. The Chime plugin cannot read mmCIF, and because Protein Explorer is built upon Chime, Protein Explorer cannot read mmCIF files. RasMol (version 2.7 or later) can read mmCIF files, but WebLab Viewer Lite, Cn3D, and DeepView cannot.

Contents of the Protein Data Bank

The Protein Data Bank is the internationally accepted repository of all published 3D macromolecular structures. Currently, there are over 15,000 PDB files available from the Protein Data Bank (PDB). This number is growing rapidly due to streamlining of structure determination methods. However, many of the entries are mutations or different experiments with the same molecule. Depending on how stringent your criteria, the number of sequence related molecules in the PDB with good resolution is only a few thousand (*see* Website: http://www.fccc.edu/research/labs/dunbrack/culledpdb.html). Less than two thousand are human proteins. Since the human proteome is encoded by approx 35,000 genes, the 3D structures that are available only represent a small percentage of human proteins. Moreover, a large percentage of the PDB files are solutions for single domains and not entire proteins.

Certain categories of proteins are underrepresented at the PDB. In most cases, the protein must be crystallized to enable its structural determination (*see* the following section). Proteins with hydrophobic surfaces, notably transmembrane proteins, tend to precipitate in an amorphous mass, rather than forming regular crystals. Therefore, transmembrane proteins are underrepresented, but several are listed in Table 2. In addition, large proteins are also underrepresented due to difficulties in crystallization. Often, single domains can be crystallized when intact proteins cannot. Nevertheless, some astounding successes have been achieved with large molecules, like the nucleosome and the entire ribosome (*see* Table 2).

Origins and Limitations of Empirical 3D Structure Data

X-ray crystallography or nuclear magnetic resonance is empirically the most reliable way to determine the 3D conformation of a macromolecule. Crystallographic results are characterized by an average resolution value given in Ångstroms. On the average, the uncertainty in the position of an atom is roughly one-fifth to one-tenth of

Table 2
Some Interesting Molecules and Their PDB Identification Codes[a]

Molecule	*Identification code*
Acetylcholinesterase with inhibitor	1VOT
Alpha hemolysin transmembrane heptamer	7AHL
Antibody (intact IgG)	1IGT
Antibody Fab bound to lysozyme	1FDL
Bacteriorhodopsin	1C8R
Calcineurin (ser/thr phosphatase), FKBP, FK506	1TCO
Calcium transporting ATPase including transmembrane	1EUL
Calmodulin (NMR ensembles)	1CFC (slight flexibility) or 2BBN (much flexibility)
Green fluorescent protein	1EMB
Hemoglobin, deoxy	1HGA
Hemoglobin, oxy	1HHO
Hemoglobin, sickle	1HBS
Hemolysin transmembrane (mushroom)	7AHL
HIV protease with inhibitor	1OHR
Lipase	1LPM vs 1TRH
Lysozyme, human	1LZR
Major Histocompatibility I (with virus peptide)	2VAB
Myoglobin, oxy	1MBO
Nucleosome (histones + DNA)	1AOI
Potassium channel	1BL8
Ribosome with tRNAs and mRNA	1GIX plus 1GIY[b]
RNA Polymerase, T7	1QLN
Transcriptional regulator, Gal4 + DNA	1D66
Transfer RNA, Phenylalanine	1TRA
Trypsin (porcine pancreatic) complexed to soybean trypsin inhibitor	1AVX
To search for other molecules *see* Website: http://www.pdblite.org	

[a] (PDB codes are case-independent: 1VOT is the same as 1vot.)

[b] The ribosome is too big to fit in a single PDB file when all side chains are included. A PDB file containing only the phosphorus atoms for the rRNA, and only the alpha carbons for proteins, is available (*see* Website: http://www.molvis.sdsc.edu/pdb/1gix1giy.pdb). This URL can be entered in the long slot on the FrontDoor of Protein Explorer.

the resolution value for high-quality data. However, uncertainty varies in different parts of the molecule as a result of the varying degrees of disorder in the crystal. In Protein Explorer's QuickViews, this can be visualized by applying the *Temperature* color scheme. Most published crystallographic results have resolutions in the range of 1.8–2.5 Å. Small values of resolution, such as 1.2 Å, mean very high resolution. Even hydrogen atoms may be resolved while large values, e.g., 5.0 Å, mean low resolution, such that only the general outlines of the backbone, but not details of side chain positions could be resolved.

Nuclear magnetic resonance (NMR) has the advantage that the protein molecules are in aqueous solution while measurements are made. In fact, the molecules need to tumble rapidly in order to achieve high resolution. This limits NMR experiments to proteins of approx 30 kD or less in size. While the result of crystallography is a single model that best fits the diffraction results, the result of a successful NMR determination is an ensemble of models all consistent with the data. Protein Explorer has excellent support for viewing such ensembles, e.g., *see* calmodulin in Table 2. About 15% of the results in the PDB are from NMR, the remainder being largely from X-ray crystallography. There are a few theoretical models in the PDB and these are much less reliable.

Protein crystals used in X-ray diffraction contain approx 50% water. Some proteins have been determined both by X-ray crystallography and by NMR, with consistant results. This gives confidence that crystallographic results reflect the conformations of proteins in aqueous solution. A more detailed introduction to the origins and limitations of macromolecular structural data is available on-line (*see* Website: http://www.rcsb.org/pdb/experimental_methods.html).

Theoretical Models of Proteins

When empirical structural data are lacking, one can turn to theoretical models. These are attempts to predict the 3D conformation from the amino acid sequence. There are two categories of theoretical models; *ab initio* models and homology models. *Ab initio* modeling attempts to predict the folded conformation of a protein using general principles of protein structure. It is moderately successful for predicting secondary structure, but poor for predicting tertiary or quaternary structure.

Homology modeling requires that an empirical structure be available for a protein with greater than 25% sequence identity. The amino acid sequence is then threaded into the template structure, following a carefully constructed sequence alignment. Homology models are more reliable than *ab initio* models at predicting the general fold of a protein, especially when the sequence identity with the template is high. However, neither method is capable of predicting side-chain positions reliably, and even homology models based on >80% sequence identity occasionally have major errors in tertiary structure. Protein Explorer includes detailed instructions on how to use existing web resources for homology modeling, notably Swiss-Model, DeepView, with links to tutorials for the latter by Gale Rhodes.

Finding Your Molecule

As explained earlier, in many cases, no empirical 3D structure is available for a molecule of interest. Nevertheless, structures are available for thousands of interesting molecules. There are many search interfaces for the Protein Data Bank, and each has its strengths and weaknesses. If you are having trouble finding a molecule, be sure to try several search interfaces. Sometimes one search interface fails to identify the relevant entries that exist in the PDB, while another succeeds.

If you have a specific molecule of interest and you have little experience searching the PDB, a good place to start is with PDB Lite. This interface is simple, avoids unnecessary jargon, explains the terms it uses, and provides detailed instructions for saving PDB files to your disk. A more sophisticated search form is provided at SearchFields (*see* Website: http://www.pdb.org) or OCA (*see* Website: http://www.bioinfo.weizmann.ac.il:8500).

If you want to browse general categories of molecules, try PDB At A Glance (*see* Website: http://www.cmm.info.nih.gov/modeling/pdb_at_a_glance.html). For further sources of PDB files; for both large and small molecules, including lipid bilayers and micelles, *see* Website: http://www.molvisindex.org.

Molecular Visualization Needs Satisfied by Protein Explorer

Ease of Use

Protein Explorer offers three tiers of interaction designed for beginners, intermediate users, and advanced users. These tiers correspond to sections in the software named *FirstView*, *QuickViews*, and *Advanced Explorer* respectively. FirstView shows a highly informative first image of the molecule together with an explanation. QuickViews provide menus and buttons that enable moieties of interest to be rendered and colored interactively to answer additional questions about the molecule. Finally, Advanced Explorer offers powerful features that require more experience to use effectively, including sections that give increased control over some visualization modes available as less flexible one-click operations in QuickViews.

Following a two-day Protein Explorer workshop, a high school biology teacher commented, “It is the most user-friendly program I have ever encountered.” The visualization tools that Protein Explorer offers you at the outset are designed for nonspecialists and people who need visualization on an occasional basis. Knowledge of basic biochemistry is assumed, but beyond that, Protein Explorer is designed to be self-explanatory. An introductory overview of its capabilities is built in as the *One-Hour Tour*. Protein Explorer’s menu system, *QuickViews*, automatically displays contextual help and access to relevant resources whenever a menu or button is clicked. When you do not know how to do something, or how best to accomplish your objective, consult the *Help/Index/Glossary*. For those planning to use Protein Explorer often, an extensive Tutorial is provided. All of these resources are accessible from Protein Explorer’s *FrontDoor* page (*see* Website: http://www.proteinexplorer.org). This remains open in a background window while one or more Protein Explorer sessions are running.

Getting Started Quickly and Smoothly

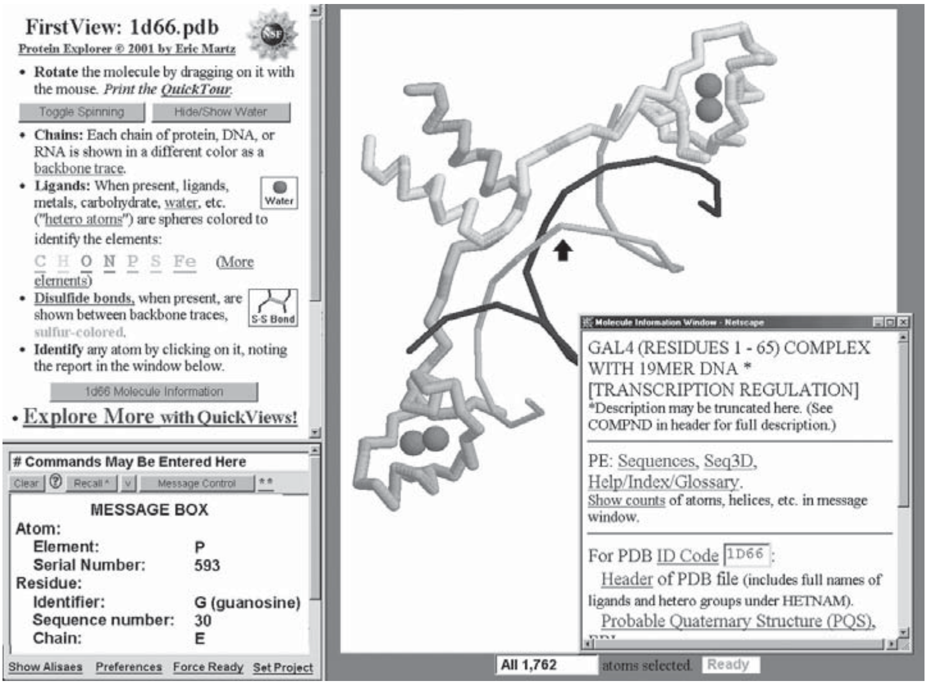
To try Protein Explorer, *see* Website: http://www.proteinexplorer.org and click on the large *Quick-Start* link near the top of the first page, which is called the *FrontDoor*. Printing and following the *One-Hour Tour* that is prominently linked on the *FrontDoor* screen is the recommended way to begin.

Protein Explorer works in the Netscape browser (Windows or Macintosh) or in the Microsoft Internet Explorer browser (Windows only). At Website: http://www.proteinexplorer.org, when you click on **QuickStart**, Protein Explorer will tell where to get the free Chime plugin if needed. It will refuse to start until Chime is installed. If you have problems getting Protein Explorer to display the *QuickStart* molecule, use the *Troubleshooting* link at the *FrontDoor*.

FirstView: Informative and Friendly

See companion CD for color Fig. 1

The first view of a molecule offered by Protein Explorer is designed to be maximally informative (*see* Fig. 1). Protein Explorer’s *FirstView* page describes this image and offers extensive background information on how protein and nucleic acid back-



FirstView: 1d66.pdb

Protein Explorer © 2001 by Eric Martz

- **Rotate** the molecule by dragging on it with the mouse. *Print the QuickTour.*
- **Chains:** Each chain of protein, DNA, or RNA is shown in a different color as a backbone trace.
- **Ligands:** When present, ligands, metals, carbohydrate, water, etc. ("hetero atoms") are spheres colored to identify the elements: C H O N P S Fe (More elements)
- **Disulfide bonds,** when present, are shown between backbone traces, sulfur-colored.
- **Identify** any atom by clicking on it, noting the report in the window below.
- Explore More with QuickViews!

Commands May Be Entered Here

MESSAGE BOX

Atom:

Element: P

Serial Number: 593

Residue:

Identifier: G (guanosine)

Sequence number: 30

Chain: E

Molecule Information Window - Netscape

GAL4 (RESIDUES 1 - 65) COMPLEX WITH 19MER DNA * [TRANSCRIPTION REGULATION]

*Description may be truncated here. (See COMPND in header for full description.)

PE: Sequences, Seq3D, Help/Index/Glossary.

Show counts of atoms, helices, etc. in message window.

For PDB ID Code 1D66:

Header of PDB file (includes full names of ligands and hetero groups under HETNAM).

Probable Quaternary Structure (PQS),

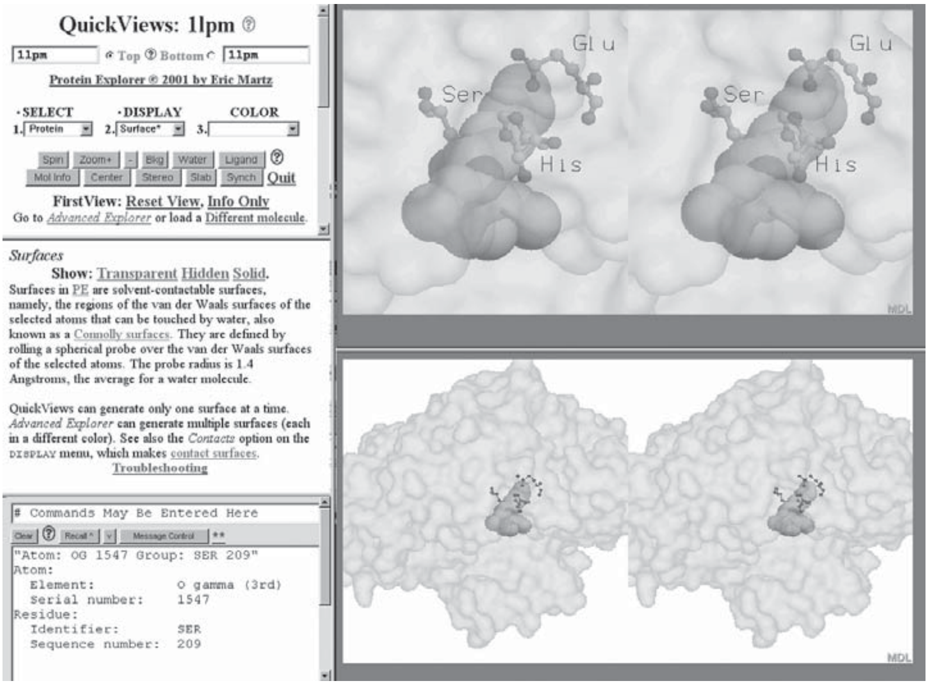
All 1,762 atoms selected.

Fig. 1. Protein Explorer's *FirstView* strives to be maximally informative. The molecule shown, PDB identification code 1D66, is the DNA-binding domain of Yeast Gal4 complexed to a 19 base pair DNA double helix. Rotating the image with the mouse makes 3D relationships clear, difficult to appreciate in this static screenshot. FirstView enables you to observe how many chains are in the molecule, which are protein and which are nucleic acid, and whether disulfide bonds or ligands (spacefilled) are present. Protein chains have been made thicker than DNA chains in this figure; normally all chains are the same thickness and can be distinguished by color. Extensive background information is available through hyperlinks on this screen. For example, backbone traces and simplified disulfide bond rendering are explained graphically (not shown). Clicking on any atom reports its identification in the message box, hence protein chains can be easily distinguished from nucleic acid chains. The arrow indicates the atom clicked for this figure. Water, currently hidden, is shown by default (configured in *Preferences*), but can be shown or hidden with the button **Hide/Show Water**. The window at the lower right can be closed or reopened at any time by pressing the **Molecule Information** button.

bone traces and disulfide bonds are rendered, on water in protein crystals, and a key to the standard Corey-Pauling-Koltun color scheme used to identify elements in ligands. You are ready to leave the *FirstView* page when you know how many chains are present in the molecule, whether each chain is protein or nucleic acid, and whether disulfide bonds or ligands are present.

QuickViews: Easy but Powerful Interactivity

The most fundamental need in any molecular visualization software is the ability to interactively render and color components of the molecule as desired. Often hiding some portions of the molecule helps to visualize the moieties of interest. Protein Explorer provides these basic visualization capabilities in its *QuickViews* menu system.



Surfaces

Show: Transparent Hidden Solid.

Surfaces in PE are solvent-contactable surfaces, namely, the regions of the van der Waals surfaces of the selected atoms that can be touched by water, also known as a Connolly surfaces. They are defined by rolling a spherical probe over the van der Waals surfaces of the selected atoms. The probe radius is 1.4 Angstroms, the average for a water molecule.

QuickViews can generate only one surface at a time. *Advanced Explorer* can generate multiple surfaces (each in a different color). See also the *Contacts* option on the DISPLAY menu, which makes contact surfaces.

Troubleshooting

```
# Commands May Be Entered Here
```

```
"Atom: OG 1547 Group: SER 209"
Atom:
  Element:          O gamma (3rd)
  Serial number:    1547
Residue:
  Identifier:       SER
  Sequence number:  209
```

Fig. 2. The *QuickViews* menu system is the heart of the user-friendly power of Protein Explorer. Select, Display, and Color menus (*see* Fig. 3) and buttons make it unnecessary to learn any RasMol-style commands. However, Rasmol commands may be entered freely by those who have learned them. Buttons include **Spin**, **Zoom**, **Background** (toggles between black and white), **Water**, **Ligand**, **Molecule Information**, **Center** (clicked atom or currently selected atoms), **Stereo**, **Slab**, and **Synch**. Each menu or button action automatically displays help in the left middle frame. Sometimes the help includes submenus. In this case, there are links to make the protein surface transparent, hidden, or solid (it is currently transparent). This image shows an inhibitor (spacefilled) bound to the catalytic site of lipase (divergent stereo pair). The catalytic triad (ball and stick) is visible inside the transparent surface of the protein. This screenshot is of *Protein Comparator*, which is identical to ProteinExplorer except that two molecules can be compared, one above the other. All menus and buttons work on one molecule at a time, selected by the **Top** — **Bottom** radio buttons. In this case, the same molecule was loaded in both top and bottom positions, but is shown at different zoom levels.

Basic Rendering, Coloring, and Hiding

See companion CD for color Fig. 2

Most rendering needs are met by QuickViews' DISPLAY menu (*see* Figs. 2–4). To keep the QuickViews menus compact to fit in the available space, short keyword identifiers are used for most menu options; however, a detailed explanation is automatically displayed in the middle *help* frame when any option is chosen. Therefore, the way to become familiar with the menu options and what they mean is to try them out. In an important enhancement that goes beyond RasMol, molecular surfaces, constructed with a rolling solvent probe, can be created quickly and displayed and rotated in real time (*see* Figs. 2 and 4).

In order to apply different renderings and color schemes to different parts of the molecule, subsets of atoms must first be selected. Protein Explorer's SELECT menu (*see* Fig. 3) offers many commonly used subsets of atoms, including each chain, and ligand. When multiple ligand molecules or chains are present, an arbitrary subset can

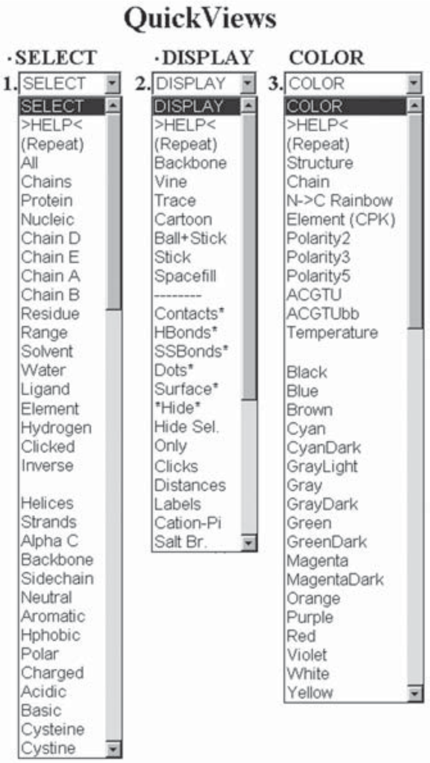


Fig. 3. QuickViews menus expanded to show all options. These menus and associated features enable sophisticated interactive visualization without learning the RasMol command language. The paradigm for modifying the image of the molecule, adopted from RasMol, is to select a subset of the atoms, display or render them as desired (or hide them), and color them if desired. Repeating these three steps on different subsets of atoms produces complex displays. Choosing **Clicked** on the SELECT menu changes the action of the mouse clicks to selection, rather than the default (atom identification). Using mouse clicks facilitates selection of arbitrary subsets of atoms, residues, or chains. (A separate window not shown, *Seq3D*, enables selecting arbitrary residues or ranges by clicking on a sequence listing.) Every menu selection automatically displays help, or for color schemes, a color key, in the left middle frame shown in Fig. 2. A section of QuickViews called *QuickViews Plus* (not shown) enables Boolean operations between items on the SELECT or DISPLAY menus (access by scrolling down in the upper left frame in Fig. 2). For example, one can select **Protein**, then subtract from the selection **Helices** and **Strands**; or one can add a ball and stick rendering to a backbone rendering. Some options on the DISPLAY menu invoke complex command scripts, notably **Contacts** (*see* Fig. 4), **Cation-Pi**, and **Salt Br**. Nevertheless, from the perspective of the user, these are one-click operations.

be selected by clicking with the mouse. The number of currently selected atoms is always shown in a slot beneath the molecular image (*see* Fig. 1).

See companion CD for color Fig. 3

A color scheme to reveal each of the following is a one-click operation from the COLOR menu (*see* Fig. 3): locating the amino and carboxy termini (with a rainbow color sequence, N->C Termini); secondary structure (alpha helices, beta strands, and turns); distribution of hydrophobic vs. polar amino acid side chains (**Polarity2**, two colors); charged residues (**Polarity5**, 5 colors); nucleotides (A, C, G, T and U); and regions of higher disorder in the crystal (temperature).

Sequences: Detecting Gaps and Identical Chains

Protein Explorer can display the sequences of the chains in your molecule. The *Sequences* and *Seq3D* (*see* following section) pages are accessed through the Molecule Information Window, available by clicking the **Mol. Info.** link that is present on all tiers of Protein Explorer. This display is especially useful for spotting whether the structure has gaps because the sequences are taken directly from the atomic coordinates. Leading and trailing ends of a chain are often unresolved in protein crystals because they are disordered. Sometimes, surface loops are missing for the same reason. These gaps are difficult to detect by looking at the 3D structure. Another use of the sequences display is to discover whether there are multiple copies of chains with identical sequences. Protein sequences are displayed in one-letter amino acid code, but touching any residue with the mouse pointer displays the corresponding three-letter code.

Sequence to 3D Mapping

In addition to an annotated full-screen sequences display, Protein Explorer offers a more compact listing of sequences that can be viewed alongside the 3D image. In this window, called *Seq3D*, clicking on a residue (or range of residues) highlights the positions in the 3D structure. It also leaves them selected, so they can be rendered and colored as desired. *Seq3D* includes the option to *scrutinize a range*. This highlights a range of amino acids so it is easy to tell whether a gap in the results from the sequence absence of residues in the structure, due to disorder or is merely a residue numbering artifact with no residues actually missing.

Measurements and Labels

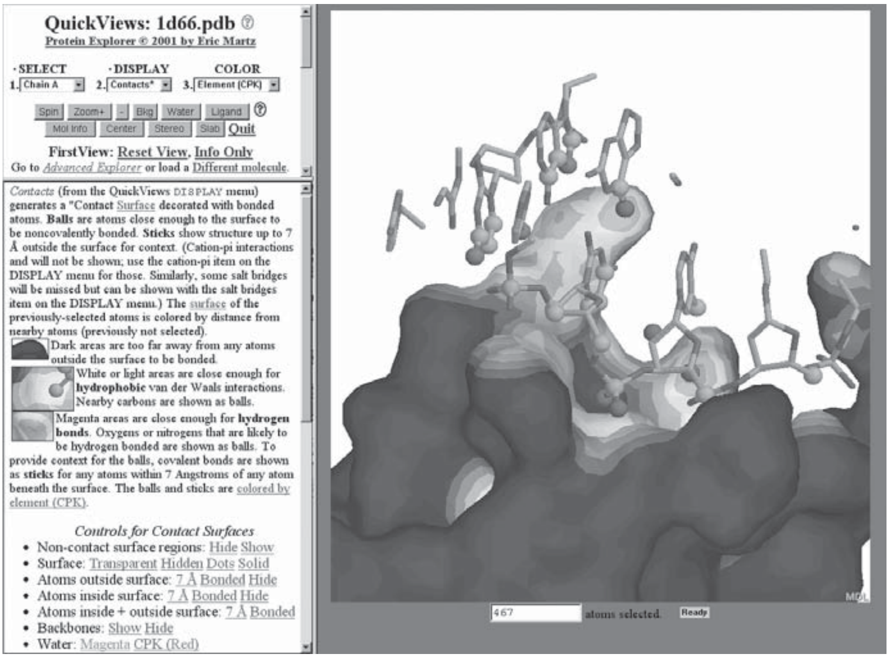
Identification labels, distances between atoms, dotted lines between atoms labeled with distances (*monitor lines*), or simple and dihedral angle values can be displayed using the **Distances**, **Labels**, or **Clicks** options of the *QuickViews* DISPLAY menu. Identification labels may be customized as in the example at top of Fig. 2, by entering arbitrary text onto a form. These can then be applied to the atoms by clicking on them. All these are done from menus and forms. Knowledge of command language syntax is not required. The current action of mouse clicks is always shown in a one-line message at the top of the middle help frame, unless it is the default messaging atom identification.

Zooming, Centering, Stereo

It is important to be able to easily enlarge (zoom) or reduce the size of the image, at will. A cluster of convenience buttons is accessible on all pages (except *FirstView*) including zoom and centering. Centering can operate on the currently selected atoms, or on a single atom designated by clicking. Centered atoms remain centered during rotation and zooming. A stereo button toggles the split image stereo on and off. This also provides a link to the help frame, for instructions for learning how to view stereo. Convergent (cross-eyed) vs divergent (wall-eyed) stereo is set with a preference checkbox.

Contact Surfaces Reveal Interfacial Bonds

Contact surface displays are one of the most unique and powerful displays offered by Protein Explorer (*see* Fig. 4). One-click on DISPLAY **Contacts** shows the atoms that are likely noncovalently bonded to the currently selected group of atoms on the contact surfaces. Interfacial bonds can be shown for an entire chain, a single secondary structure element (helix or strand), a ligand molecule, a single residue, a single



1. Chain A
2. Contacts*
3. Element (CPK)

Go to *Advanced Explorer* or load a Different molecule.

Contacts (from the QuickViews DISPLAY menu) generates a "Contact Surface decorated with bonded atoms. **Balls** are atoms close enough to the surface to be noncovalently bonded. **Sticks** show structure up to 7 Å outside the surface for context. (Cation-pi interactions and will not be shown; use the cation-pi item on the DISPLAY menu for those. Similarly, some salt bridges will be missed but can be shown with the salt bridges item on the DISPLAY menu.) The surface of the previously-selected atoms is colored by distance from nearby atoms (previously not selected).

Dark areas are too far away from any atoms outside the surface to be bonded.

White or light areas are close enough for **hydrophobic** van der Waals interactions. Nearby carbons are shown as balls.

Magenta areas are close enough for **hydrogen bonds**. Oxygens or nitrogens that are likely to be hydrogen bonded are shown as balls. To provide context for the balls, covalent bonds are shown as **sticks** for any atoms within 7 Angstroms of any atom beneath the surface. The balls and sticks are colored by element (CPK).

Controls for Contact Surfaces

- Non-contact surface regions: Hide Show
- Surface: Transparent Hidden Dots Solid
- Atoms outside surface: 7 Å Bonded Hide
- Atoms inside surface: 7 Å Bonded Hide
- Atoms inside + outside surface: 7 Å Bonded
- Backbones: Show Hide
- Water: Magenta CPK (Red)

467 atoms selected. Ready

Fig. 4. The *Contacts* surface display. Two clicks reveal noncovalent bonding relationships across any interface. The user selects the moiety to be surfaced, then DISPLAY **Contacts**. In this case, **Chain A** (one of the two homodimer protein chains in the DNA-binding domain of Gal4, 1D66) was selected, and is shown as a surface. The surface is rendered and automatically decorated with small spheres (*balls*), which represent atoms close enough to appropriate atoms beneath the surface to be noncovalently bonded. Sticks are added to the balls to show all covalent bonds up to 7 Ångstroms from the surface. All portions of the structures outside the surface that are farther than 7 Å from the surface are hidden. (The DNA backbone can be shown for context using **Backbones: Show** in the lower left frame.) The surface is colored by distance from the nearest atoms. Dark areas of the surface are too distant from any atoms to be noncovalently bonded. Light areas are at distances suitable for van der Waals interactions, and magenta areas (*see* companion CD for color) are close enough for hydrogen bonds or salt bridges. The *finger* of this member of zinc-finger superfamily can be seen protruding upwards, where it contacts several DNA bases in the major groove. In color (*see* companion CD), it is immediately apparent that the noncovalently bonded atoms (balls) include oxygens and nitrogens, hence that the protein is recognizing a specific base sequence. Clicking on the nucleotide base rings that include balls, from left to right (in the rearmost chain D), reports CGG, the published recognition sequence for Gal4.

See companion CD for color Fig. 4

atom, or any other selected subset of atoms. The surface of the selected subset of atoms is shown, decorated with the atoms noncovalently bonded to it (*see* Fig. 4). Putatively noncovalently bonded atoms, shown as balls, are defined as oxygen or nitrogen atoms within 3.5 Å of an oxygen or nitrogen beneath the surface and carbon or sulfur atoms within 4.5 Å of a carbon or sulfur beneath the surface. After displaying a contact surface, the middle help frame offers a submenu with many options for modifying the initial image. Such modifications include hiding noncontacted regions of the surface, rendering the surface transparent and showing the likely noncovalently bonded atoms beneath the surface. Together with centering and zooming, these make it easy to visualize and identify noncovalent bonding interactions.

Cation-Pi Orbital Interactions and Salt Bridges

Interactions between cations and the faces of aromatic rings (pi orbitals) typically make an energetically significant contribution to the stability of a folded protein. They can be displayed in one click. At the time of this writing, they are identified with approximate distance-based criteria. Energetically significant cation-pi interactions can be confirmed by consulting Gallivan and Doherty's CaPTURE server, which can be accessed through the link automatically shown in the middle help frame in *QuickViews*. It is planned that a future release of Protein Explorer will display only the energetically significant cation-pi interactions, obtained directly on-line from the CaPTURE server. Anions and cations within 4.0 Å of each other may be highlighted as likely salt bridges with a one-click operation (DISPLAY Salt Bridges). *Advanced Explorer* enables ions or aromatic rings within ligands to be included in the cation-pi and salt bridge displays.

Advanced Explorer

The Advanced Explorer section within Protein Explorer offers increased flexibility for some operations available in *QuickViews*, and some additional capabilities discussed below. An example of increased flexibility is the option to include aromatic rings of ligands, or charged atoms, in the displays of cation-pi interactions or salt bridges and to adjust the distances used to identify and highlight these interactions. Another section offers more control over construction of contact surface displays than is available in QuickViews. The *Surfaces* section of Advanced Explorer enables multiple surfaces to be concurrently shown. This enables the probe radius to be adjusted and offers molecular electrostatic potential and molecular lipophilicity potential color schemes.

The *Noncovalent Bond Finder* (NCBF) is an alternative to the *Contact Surface* display of *QuickViews*. Whereas a *Contact Surface* shows an overview of all noncovalent interactions at an interface, the NCBF allows a more detailed exploration of noncovalent bonding, better suited to large interfaces. It displays the atoms closest to any selected moiety, stepping out in 0.1 Å shells with each click on the **Find** button. To simplify the view, the display can be limited to subsets of noncovalent bonds, such as only hydrophobic, or only hydrophilic bonds.

Animation and NMR Ensembles

Multiple models of the same molecule may be contained in a single PDB file. NMR experiments result in an ensemble of alternative models, all consistent with the data (*see* Website: http://www.rcs.org/pdblexperimental_methods.htm). *Advanced Explorer* has an *NMR Models* section with numerous buttons and form slots to facilitate visualization of the models in an NMR ensemble. NMR ensembles can also be animated, simulating thermal motion.

Two different conformations of a protein are sometimes observed experimentally. When there are large differences between the conformations, animation of the transition with a *morph* helps the eye to visualize the changes in specific regions. These morphs include at least a dozen intermediate models interpolated between the experimental models. *Advanced Explorer's Animation* control page enables such morphs to be shown as *movies*. However, such animations differ from true movies in being able to be played from any rotational perspective or magnification on an arbitrary center. Moreover, these animations can be played in a variety of renderings, e.g., backbone,

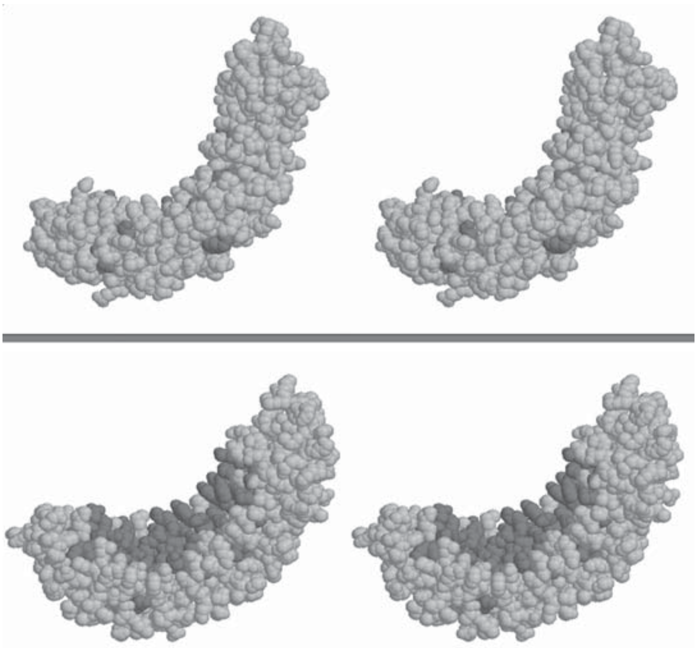


Fig. 5. Conserved regions (dark) of the surface of the PUF translation repressor Pumilio 1 (1LB2) revealed by Protein Explorer's MSA3D (divergent stereo). There is little conservation on the convex surface (top), but a large conserved patch in the concave surface of the same molecule (bottom) is believed to bind to the 3′ untranslated region of the mRNA. A multiple protein sequence alignment was constructed (outside of Protein Explorer) that included sequences from human, *Xenopus*, *Drosophila*, Yeast (2), *Dictylostelium*, and *C. elegans*. A 3D crystallographic structure is available for the human sequence (1LB2). The sequence alignment (in FASTA or PIR format) was pasted into a form box in **MSA3D**, the consensus requirement set to 80% for the seven aligned sequences. With one click of a button, **MSA3D** produced a text listing of the alignment (not shown) assigning three colors representing identity, similarity, and difference. The colors were then automatically applied to the 3D molecular model. (Thanks to David S. Bernstein for providing this alignment and example.)

cartoon, wireframe, spacefilled, and color schemes. By clicking on the animated icon at the upper left of Protein Explorer's FrontDoor, the conformational change of a protein EF hand upon binding calcium will be presented.

Coloring by Multiple Sequence Alignment (MSA3D)

See companion CD for color Fig. 5

Multiple protein sequence alignments reveal patterns of conservation or mutation during evolution. *Advanced Explorer's MSA3D* feature assigns colors to the alignment, representing conservation or mutation, and then applies these colors to the 3D image (*see* Fig. 5). Conserved regions in the 3D structure can then be visualized. The requisite sequence alignments must be prepared outside of Protein Explorer with appropriate websites or software. MSA3D includes ready-made built-in examples and a tutorial with instructions for preparing an alignment at the *Biology Workbench* (*see* Website: http://workbench.sdsc.edu).

WWW

Side by Side Comparisons

Protein Comparator is available from the FrontDoor of Protein Explorer. It is a full-featured mode of Protein Explorer that shows two molecules side by side. In Windows, rotations with the mouse can be synchronized so that both molecules rotate together when one is dragged.

Molecular Visualization Needs Not Satisfied by Protein Explorer

Limitations to Visualization

Protein Explorer's ability to display hydrogen bonds is inadequate. Currently, it can only display protein, backbone-to-backbone hydrogen bonds, and in DNA or RNA, only canonical Watson-Crick base-pair hydrogen bonds. Thus, it cannot automatically display nonstandard hydrogen bonds in nucleotides, nor hydrogen bonds in proteins that involve solvent or side chains. Better visualization of hydrogen bonds is possible in DeepView. Protein Explorer is not the best option for publication quality images, although in some cases it can suffice. This results from a compromise between image quality and speed of rendering, enabling Protein Explorer's images to be rotated smoothly in real time on ordinary personal computers. This is a feature of the Chime plugin and is unlikely to be changed. Among freeware, WebLab Viewer Lite easily produces images suitable for publication. However, interactive rotation of its high quality images is much slower, making depth perception problematic unless images are viewed in split image stereo. But unlike Protein Explorer, WebLab Viewer Lite can show alpha helices as cylinders, or nucleotide bases as cartoons.

No Modeling

Protein Explorer is limited to visualization; it cannot do modeling. Modeling, in the strict sense, means building models, or making chemical or conformational changes in existing models. Covalent bonds cannot be formed nor broken in Protein Explorer, nor can atoms be moved relative to each other. Thus, Protein Explorer cannot do *docking*, that is, for example, it cannot move a ligand into a favorable binding position on a protein. It cannot mutate amino acid side chains, or do energy minimization. However, DeepView is the only freeware package capable of substantial modeling.

Comparisons with Other Freeware Visualization Packages

Chime and RasMol

As explained in the section Protein Explorer vs Chime and RasMol, because of its substantially greater ease of use and power, Protein Explorer is preferable to Chime (by itself) or RasMol for most common visualization purposes. For a detailed comparison *see* Website: http://molvis.sdsc.edu/protexpl/pe_v_ras.htm

Cn3D from NCBI

Cn3D is the visualization freeware offered by the US National Center for Biotechnology Information (NCBI; *see* Weblinks). Its main goals include showing the user an informative first view of a molecule (an excellent feature which was released in Cn3D well before Protein Explorer was designed), and enabling interactive exploration, similar to Protein Explorer. It is a well-developed package with many strong features. It works on unix as well as Windows and Macintosh platforms, and has open source code. Cn3D accepts atomic coordinate files only in the ASN.1 format peculiar to the NCBI; it cannot read PDB format. The advantage is that the ASN.1 atomic coordinate files have better internal consistency and fewer errors than do PDB files; one disadvantage is that there is a lag between when a PDB file is first released, and when it becomes available in ASN.1 format.

Cn3D is, in my opinion, not as easy to use as Protein Explorer, especially for beginners. It has no context-sensitive help (in contrast with QuickViews in Protein Explorer). Cn3D has no presentation capabilities; such capabilities are under development for Protein Explorer at the time of this writing (*see* Website: http://www.umass.edu/microbio/chime/pipe).

Cn3D's image quality is higher than Protein Explorer's, and as a consequence, rotation of the molecule is slower and jumpier. Its sequence-to-structure interface is superior to Protein Explorer's in that it is clickable in both directions (sequence to structure and structure to sequence) and it maintains the same colors in both the sequence and structure displays. Rendering and coloring operations are generally at the residue level; it is less able to operate at the atomic level than is Protein Explorer. It can render helices as Cylinders (unlike Protein Explorer), but cannot render surfaces (Protein Explorer can). All views are customized from menus; there is no scripting or command language. A file can easily be saved that will restore the same view of a molecule in a later session; this is more difficult to do in Protein Explorer. Many PDB files give alternative conformations for amino acid side chains; Cn3D handles these very well, while Protein Explorer handles them badly. Cn3D's support for coloring a 3D structure according to a multiple sequence alignment appears to be more fully developed (more color schemes) than Protein Explorer's, but more difficult to use. Cn3D lacks the kind of generalized animation capability present in Protein Explorer, and has nothing equivalent to the Contact Surfaces, Noncovalent Bond Finder, Cation-Pi, or Salt-Bridge displays in Protein Explorer.

WebLab Viewer Lite

WebLab Viewer Lite is the free version of the commercial WebLab Viewer Pro from Accelrys Inc. (formerly Molecular Simulations, Inc.). Both Pro and Lite versions are available for Windows; Lite is available for Macintosh. The Pro version has extensive modeling capabilities lacking in the Lite version (which can however add hydrogen atoms, and change torsion angles). Image quality is higher in WebLab than in Protein Explorer, with the consequence that images rotate much more slowly, making it difficult to perceive depth without using split-image stereo. Because of the high image quality, WebLab is especially suited to printing publication-quality images. It can simplify alpha helices to cylinders (unlike Protein Explorer). In addition to the usual renderings, WebLab can show surfaces, and color them by molecular electrostatic potential (MEP). Its MEP-colored surfaces are strikingly better-looking than those of Protein Explorer (*see* Website: molvis.sdsc.edu/protexpl/mep.htm).

WebLab has some unusual features. It can display all hydrogen bonds, label centers of chirality with "R" or "S", and show "bumps" (atoms that are too close, namely less than 70% of the sum of their covalent radii). It has a Crystal Builder that fills out crystallographic unit cells and neighbors (but this is a great deal slower than the comparable feature in DeepView).

In my opinion, WebLab Lite's ease of use is not as great as that of Protein Explorer, but is fair. There is no context sensitive help, and color keys do not appear for the numerous color schemes available. There is no tutorial, but there is an extensive reference manual (in the form of a Windows help file).

Modeling with DeepView Freeware

DeepView, also known as SwissPDB-Viewer, is a powerful modeling and visualization package for Windows, Macintosh, SGI and linux. It is a closed-source freeware program from GlaxoWellcome. For purely visualization purposes, its images are not as attractive or compelling as those of Protein Explorer, Cn3D, or WebLab. Many useful components of the images disappear during rotation, making stereo viewing especially important. DeepView's user-friendliness is at the low end of the scale, but Gale Rhodes has written a number of excellent tutorials for it (accessible through the DeepView section of molvisindex.org), and procedures for mutating, structural alignment, or building crystal contacts in DeepView are built into Protein Explorer (see below).

DeepView's modeling capabilities are powerful and extensive, and include energy minimization, and integration with the SWISS-MODEL web server for homology modeling. A comprehensive overview of DeepView's capabilities is beyond the scope of this chapter. However, DeepView's capabilities complement the visualization capabilities of Protein Explorer nicely, and below are summarized some of the most popular uses of DeepView for producing models to be visualized in Protein Explorer.

As mentioned above, if no empirical structure is available for a protein, the best way to visualize its 3D structure is with a **homology model**. Construction of a homology model requires that an empirical template structure be available with at least 25% sequence identity. Homology modeling is beautifully automated by SWISS-MODEL working in concert with DeepView. Protein Explorer provides orientation to homology modeling using SWISS-MODEL and DeepView (look for "homology modeling" in the Help/Index/Glossary).

People using Protein Explorer often want to know how to **mutate** selected residues in a published PDB structure. DeepView makes this very easy (look for "mutation" in Protein Explorer's Help/Index/Glossary; *see* the Major Websites box above). It is sometimes important to examine contacts between molecules in a crystal, to evaluate whether they have affected the conformations of surface features. **Crystal contacts** can be constructed using DeepView for visualization in Protein Explorer. For instructions, in Protein Explorer, open the Molecule Information Window and click on **Crystal Contacts**. Finally, DeepView provides the easiest way to create structural alignments of two or more molecules, even with multiple chains, while retaining hetero atoms. For instructions, look up "alignments" in Protein Explorer's Help/Index/Glossary.

Challenges to Help You Learn Protein Explorer

Do the QuickTour

See Website: http://proteinexplorer.org and take a careful look at the *FrontDoor* page. It is important to know about the resources that can be accessed here. Before you can begin to use Protein Explorer effectively, you will need to spend an hour or two completing the *QuickTour*. Go to proteinexplorer.org and click on the large link to the *QuickTour*. Once you have finish the *QuickTour*, it will be very worthwhile to continue into the section entitled *Beyond the QuickTour*.

Generic Questions for Any Molecule

Using a molecule of your choice from the *Interesting Molecules* from Table 2 or a search, (e.g., *see* Website: http://www.pdblite.org), go to Website: http://www.proteinexplorer.org and enter the PDB identification code in the slot at the *FrontDoor* of Protein Explorer. A good place to start is the set of questions applicable to any molecule available on-line as *Discovery in Protein Explorer* (*see* Website: http://molvis.sdsc.edu/protexpl/discover.htm, linked under *Lesson Plans*). It is important to have completed the *QuickTour* before you attempt to answer these questions. You may wish to print the questions and record your answers as a record of your progress. You will learn the most if you try to answer these questions before you look at the hints on how to answer them in Protein Explorer (*see* Website: http://molvis.sdsc.edu/protexpl/genhints.htm).

Heme and Water in Myoglobin

Myoglobin is a protein within muscle cells that stores and transports oxygen for use in oxidative metabolism. Oxygen is bound by an embedded heme group, identical to that found in the hemoglobin of red blood cells. Heme gives myoglobin a reddish-brown color. The muscles of animals that need to store unusually large amounts of oxygen, such as marine mammals, are brown due to the large amounts of myoglobin present. Myoglobin was the first protein for which a 3D structure was obtained by X-ray crystallography, in work done by John Kendrew and coworkers in the 1950's.

Assuming you have completed the *QuickTour* and are familiar with the techniques learned, load **1MBO** into Protein Explorer. Use the generic questions (linked under *Lesson Plans*) to become familiar with this molecule. You may wish to print the questions and record your answers as a record of your progress. Myoglobin-specific questions follow.

Speed Up Performance by Saving Your PDB File to Disk

If you have a slow connection to the Internet, it will help to have a copy of the myoglobin PDB file on your local disk. After loading 1MBO from the Internet once, while it is visible in Protein Explorer, click on **MDL** to the lower right of the molecule, then use the menu that pops up for *File, Save Molecule As*, naming the file **1mbo.pdb**. You may want to create a new folder named **PDB Files** to put it in. To speed up the performance of Protein Explorer, return to the *FrontDoor*, and start **Bare Protein Explorer**. Use the Browse button to find and load **1mbo.pdb** from your disk. As a result, using **Reset View** (which reloads the molecule) will be much faster. Also, when you start another session of Bare Explorer, 1mbo.pdb will be on the menu of recently loaded molecules.

You can further speed up performance by downloading Protein Explorer itself, using the **download** link on the FrontDoor.

Myoglobin-Specific Questions

Is heme an amphipathic molecule?

That means: Does it have a largely hydrophobic (nonpolar, carbon) end and a more hydrophilic (polar) end? In QuickViews, SELECT **Ligand**, DISPLAY **Only**. (Use the **Center** and **Zoom** buttons.) (Spacefill the heme, and color it by element.)

How many iron atoms are in one heme molecule?
What holds them in place?

DISPLAY **Ball + Stick**. Protein Explorer does not show the bonds that hold the iron in place, but with a basic knowledge of chemistry you can deduce that the atoms closest to the iron have lone electron pairs facing the electron-hungry metal ion (Fe^{+++}). The molecular oxygen, O_2, bound to the iron is not covalently bound and can be released. (Bonds shown by Protein Explorer are often incorrect around metals due to limitations in Chime.)

Is any portion of the heme in contact with water?

Examine the surface of the protein. In QuickViews, use **FirstView: Reset View**. Return to QuickViews, then SELECT **Protein**, DISPLAY **Surface**. Use the **Water** toggle button to examine the protein surface with water hidden, then with water showing.

How much water is inside the protein?

Click **FirstView: Reset View**, return to QuickViews, DISPLAY **Spacefill** (for all atoms). Color water green to distinguish water oxygens from nonwater oxygens. SELECT **Water**, COLOR **Green**. Press the **Slab** button. Rotate the molecule, looking carefully for water that is buried in the inside of the protein.

What kinds of protein atoms contact the heme?

Click **FirstView: Reset View**. In QuickViews, SELECT **Ligand**, then DISPLAY **Ball + Stick**. Now, DISPLAY **Contacts**, using the step by step option. After you click the 5th checkbox (**Show likely noncovalently-bonded atoms as balls**), zoom in and notice the elements that are noncovalently bonded, and their distribution. Click checkbox 2 (**Show surface as opaque**) to uncheck it, making the surface transparent again so you can see the relations between the bonded elements and the heme elements.

How many polar protein atoms contact the heme?

Click the 8th checkbox (skipping the 6th and 7th checkboxes). Scroll down in the middle frame until you see the *Controls for Contact Surfaces*. At the bottom of the block of control links, click **Water: Magenta**.

Now you can tell water oxygens from protein oxygens. To count the contacting protein atoms, enter **select protein and balled** in the command entry slot (pressing the Enter/Return key). The report of the number of atoms selected appears in the message box so that the atoms display as small spheres.) To verify that this command selected the correct atoms, COLOR **Black**. To restore the color scheme, COLOR **Element** (CPK). To confirm the number of protein carbon atoms contacting the heme, enter the command **select protein and balled and carbon**.

Lipase

One type of lipase cleaves fatty acids from triacylglycerol. Such lipases are used both in digestion of food and for harvesting of fatty acids in fat cells, i.e., adipocytes. They are also used by microbes, a fungus *Candidia rugosa*. Load the lipase **1LPM** into Protein Explorer. Complete the QuickTour, and answer the generic questions (linked under “Lesson Plans”) before proceeding.

Lipase-Specific Questions

Is the entire surface of lipase polar? How does this relate to the function of this enzyme?

Is this lipase monomeric or oligomeric?

Open the Molecule Information Window and check the report at the Probable Quaternary Structures (PQS) website. (If you need help, click on the circled green question mark at the top, to access the Help/Index/Glossary.) When you have answered the question, close the PQS windows, returning to the Polarity2-colored lipase.

Is water excluded from the hydrophobic surface patch?

Use the **Water** button to display water. Do you see any bound to hydrophobic areas? If you do, try the Polarity3 color scheme to find out if such water is in fact bound to the polar backbone within stretches of hydrophobic side chains. You may also try Polarity5 to see whether charges occur in the hydrophobic patch. Then return to Polarity2.

Where is the catalytic site (Part 1)?

SELECT **Ligand**, COLOR **Green**. (Click the **Ligand** button if nothing green is visible.) Four green items are on the surface. More interestingly, one green molecule is centered in the hydrophobic patch, in a depression. Hide it by clicking the **Ligand** button, and you will see that it is inserted into a tunnel extending deep into the core of the protein. This molecule is an inhibitor of the enzyme that binds to the catalytic site. It has a fatty acid-like tail, but a phosphonate where the ester bond would be. What does “MPA” stand for? (Hint: consult the HET records in the PDB file header.)

Where is the catalytic site (Part 2)?

With protein selected, DISPLAY **Hide Selected**, so the protein disappears. Now **DISPLAY, Surface**, and make it transparent. With the green inhibitor showing, you can see how it penetrates into the core of the protein. Lipases are serine esterases with a catalytic triad of amino acids, in this case, Ser209, Glu341, and His449. Use Seq3D to highlight these three residues. (Help finding or using Seq3D can be found in the Help/Index/Glossary.)

Which atom of which member of the catalytic triad is covalently bonded to the phosphonate (based on a distance of less than 1.6 Å)?

SELECT **Ligand**, COLOR **Element**. Because the enzyme cannot hydrolyze the inhibitor, it cannot be released. (This enabled it to be crystallized in the active site!)

Are the residues of the catalytic triad in alpha helices or beta strands?

Interestingly, some of the crystals of this lipase formed in a *closed* conformation, with the hydrophobic patch and catalytic site obscured, e.g., 1TRH. Thus, the enzyme may be closed when soluble, and may open to engage a fat droplet at the catalytic site. At Protein Explorer’s FrontDoor, click on the small animated EF hand image at the top left. On this page about Animations, one of the examples near the bottom is a *morph* from the closed form of lipase to the open form. To look at it, click on the link to Lipase on this Animations page.

Identifying Conserved Regions

Choose a molecule of interest and follow the tutorial in Protein Explorer's MSA3D to construct a multiple protein sequence alignment using Biology Workbench. Use the resulting alignment in MSA3D to identify conserved regions of your protein. Note: This project will take several hours to complete.

Answers to Questions

Myoglobin file 1MBO is a single protein chain of 153 amino acids with no gaps. Bound to the protein are 334 water molecules, one heme molecule to which is bound molecular oxygen and one sulfate ion. The protein has no disulfide bonds and no cysteine residues. The other 19 amino acids are present. The secondary structure is all alpha, with most helices contacting the heme. The amino and carboxy termini are on the surface, and over 20 Å apart. The protein surface is largely polar and the core is entirely hydrophobic except for two histidines that contact the iron of the heme. The surface appears to have more positively charged amino acids than negatively charged. This is confirmed by the isoelectric point of 9.3 (charge of +3 at pH 7.0).

Myoglobin-Specific Questions

Heme is an amphipathic molecule because one end is made entirely of carbon, while the other end has two organic acid groups. Each heme molecule contains a single iron atom (Fe^{+++}), held into place by the lone electron pairs of four nitrogens. The hydrophobic end of heme is buried in the hydrophobic protein core, while the hydrophilic carboxyls protrude from the protein surface and contact water. There are only a few water molecules buried in the protein. For example, HOH305 is bound to the oxygen molecule that is bound to the heme iron. The vast majority (51 of 56) of the protein atoms that contact heme are carbons that form van der Waals interactions with the carbon predominant in heme; there are also one oxygen and four nitrogens, three of which are in separate histidine side-chains.

Lipase

Triacylglycerol hydrolase 1LPM consists of a single protein chain of 534 amino acids with no gaps. There are two disulfide bridges, plus one cysteine. There are two carbohydrate adducts on the surface consisting entirely of NAG (N-acetyl-D-glucosamine). They are covalently linked to Asn 314 and 351. These linkages can be seen by displaying the ligand contact surfaces. Although the covalent bonds between the Asn nitrogen and sugar carbon are not shown due to a technicality, their positions can be confirmed by the inter-atomic distances of less than 1.5 Å. Two calcium ions are bound to the protein surface, one to the side chain of Asp260 and one to the oxygen of Gly326. Of greatest interest, one ligand molecule is buried deeply within the protein. It is the inhibitor, a substrate analog, menthyl hexyl phosphonate. The protein has an alpha plus beta secondary structure, with most beta strands in a single large beta sheet. The amino and carboxy termini are on the surface at opposite sides of the protein. The carboxy terminus is near the inhibitory ligand (5.4 Å). The core of the protein is largely hydrophobic, but includes a number of polar and even charged side chains. Consistent with this, there are quite a few water molecules distributed about the interior. The protein has a largely polar surface, except for a large hydrophobic pocket, in the bottom of which sits the inhibitor. The protein surface appears to have more negatively charged side chains than positively charged. This is confirmed by the pI of 4.5, with a charge of -19 at pH 7.0.

Lipase-specific questions:

The hydrophobic surface pocket fits with function, because the substrate (fat) is hydrophobic and needs a surface region suitable for binding. Probable Quaternary Structures confirms that lipase is monomeric. The hydrophobic surface pocket has no water bound (except to protein backbone), and no charged side chains. The surface ligands were identified earlier, as was MPA. The side-chain oxygen of Ser209 is covalently bound to the phosphonate of the inhibitor. The catalytic triad residues are in neither helices nor strands. The catalytic site is centered at the edge of the large beta sheet, surrounded on the other side by loops without secondary structure, outside of which are alpha helices.

Acknowledgments

I am grateful to the Division of Undergraduate Education of the National Science Foundation for grant support, and to Joel Sussman and Philip Bourne for encouragement. Thanks to Roger Sayle for RasMol, and to Tim Maffett, Bryan van Vliet, Jean Holt, and others at MDL for Chime, without which PE would not have been possible. Thanks to Frieda Reichsman for a critical reading of the manuscript and many clarifying suggestions, and to Byron Rubin for introducing me to lipase.

Suggested Readings

Protein Explorer vs. Chime and RasMol

Bernstein, H. J. (2000) Recent changes to RasMol, recombining the variants, Trends Biochem. Sci. 25, 453–455.

Sayle, R. A. and Milner-White, E. J. (1995) RASMOL: Biomolecular graphics for all, Trends Biochem. Sci. 20, 374–376.

Contents of the Protein Data Bank

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000) The Protein Data Bank, Nucleic Acids Res. 28, 235–242.

Origins and Limitations of Empirical 3D Structure Data

Branden, C. and Tooze, J. (1999) Introduction to Protein Structure, 2nd ed., Garland Publishers, New York, NY.

Rhodes, G. (1999) Crystallography Made Crystal Clear, Academic Press, New York, NY.

Cation-Pi Orbital Interactions and Salt Bridges

Gallivan, J. P. and Dougherty, D. A. (1999) Cation-pi interactions in structural biology, Proc. Nat'l. Acad. Sci. USA, 96, 9459.

Cn3D from NCBI

Wang, Y., Geer, L. Y, Chappey, C., Kans, J. A., and Bryant, S. H. (2000) Cn3D: sequence and structure views for Entrez, Trends Biochem. Sci. 25, 300. *See* Website: http://www.ncbi.nlm.nih.gov/Structure/CN3D/cn3d.shtml.

Modeling with DeepView Freeware

Guex, N. and Peitsch, M. C. (1997) SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling, Electrophoresis 18, 2714.
See Website: http://www.expasy.ch/spdbv

Lipase

Rubin, B. (1994) Grease pit chemistry exposed, Structural Biol. 1, 568.

33 Multiple Sequence Alignment and Analysis

The SeqLab Interface: A Practical Guide

Steven M. Thompson

Introduction

SeqLab is based on Steve Smith's (1994) GDE (Genetic Data Environment) and is a part of Accelrys' Genetics Computer Group Wisconsin Package. This comprehensive package of sequence analysis programs is used worldwide and is an *industry-standard*. The GCG Package only operates runs on server computers running the UNIX operating system but it can be accessed from any networked terminal. Specialized *X-server* graphics communications software is required to display GCG's SeqLab GUI. X server emulation software needs to be installed separately on personal style Microsoft Windows/Intel or Macintosh machines but genuine X Windowing comes standard with most UNIX operating systems. *Wintel* machines are often set up with either XWin32 or eXceed to provide this function; Macintoshes are often loaded with either MacX or eXodus software.

The details of X and of connecting to a GCG server are briefly described in Chapters 13 and 17. They are machine-specific, however, there are a few useful hints that are common to all machines. X-windows are only active when the mouse cursor is in that window and buttons are turned on when they are pushed in and shaded. Rather than holding mouse buttons down, to activate items, just click on them. Do not close windows with the X-server software's close icon in the upper right- or left-hand window corner, rather, always use GCG's **Close** or **Cancel** or **OK** button located near the bottom of the window.

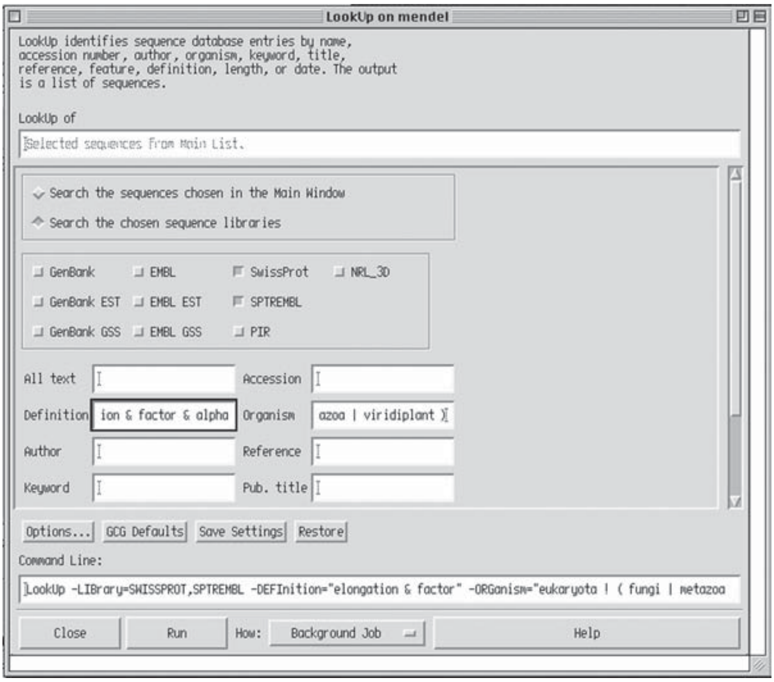
After logging on to your UNIX GCG user account, issue the command **gcg** to initialize the software suite. This initialization process activates all of the programs within the package and displays the current version of both the software and all of its accompanying databases. Issue the command **seqlab &** in your terminal window to start the SeqLab interface. The ampersand, **&**, instructs the command to launch SeqLab as a background process so that you can retain control of your initial terminal window. This should produce two new windows, the first, an introduction with an **OK** box; check **OK.** You should now be in SeqLab's List mode. Before beginning any analyses, go to the **Options** menu and select **Preferences . . .** A few of the options should be verified to insure that SeqLab runs in its most intuitive manner. The defaults are usually fine, but changes can be made. Remember, buttons are turned on when they are pushed in and shaded.

First notice that three different **Preferences** settings can be changed: **General, Output,** and **Fonts;** start with **General.** The **Working Dir . . .** setting will be the directory from which SeqLab was initially launched. This is where all SeqLab's working files will be stored. It can be changed in your accounts if desired. Be sure that the **Start SeqLab in:** choice has **Main List** selected and that **Close the window** is selected under the **After I push the Run button:** choice. Next select the **Output** Preference. Be sure that **Automatically display new output** is selected. Finally, take a look at the **Fonts** menu. If you are analyzing very large alignments, then selecting a smaller Editor font point size may be desirable in order to view more of your alignment on the screen at once. Click **OK** to accept any changes.

LookUp a Protein in the Database

Given interest in a particular biological molecular sequence, you can use any text string searching tool, such as NCBI's Network Entrez or EMBL's SRS on the World Wide Web, to find that entry's name in a sequence database. GCG's LookUp, an SRS derivative, was used to retrieve the Elongation Factor 1α dataset. After an entry has been identified, the next step is to use a sequence similarity searching program to help prepare a list of sequences to be aligned. One of the challenges in creating a multiple alignment is knowing what sequences should align. Any list from any program should be restricted to only those sequences that actually should be aligned. To generate a meaningful alignment, make sure that the group of sequences that you align belong to the same gene or related gene family.

To use entries of interest in the GCG sequence databases we need to know their proper database names or accession codes. There are several methods. The NCBI Entrez program either over the Web or installed locally in the network client/server mode is one of the more powerful. GCG's LookUp program can also be used, creating an output file that can be used as an input list file to other GCG programs. To start be sure that the **Mode: Main List** choice is selected in your main window and then launch **LookUp** through the **Functions Database Reference Searching** menu. In the new **LookUp** window be sure that **Search the chosen sequence libraries** is checked and then select **SwissProt** as well as **SPTREMBL** for the libraries to search. The representative set of elongation factor entries are all from *primitive* Eukaryotes, i.e., any Eukaryote excluding Fungi, Metazoans, and true Plants. Therefore, under the main query section of the window, type the words and symbols **elongation & factor & alpha** following the category **Definition** and the words and symbols **eukaryota ! (fungi | metazoa | viridiplantae)** in the **Organism** category; next press the **Run** button. You need to use the Boolean operator symbols to connect the individual query strings because the databases are indexed using individual words for most fields. The **Organism** field is an exception, it will accept *Genus species* designations as well as any other single word supported level of taxonomy, e.g., *fungi.* The Boolean operators supported by LookUp are the ampersand, **&**, meaning *AND,* the pipe symbol, **|** , to denote the logical *OR,* and the exclamation point, **!** , to specify *BUT NOT.* Other LookUp query construction rules are case insensitivity, parenthesis nesting, * and ? wildcard support, and automatic wildcard extension. This query should find most of the elongation factor alpha's from the *lower* eukaryotes in the SwissProt and SPTREMBL databases and will provide a reasonable and interesting starting dataset for the chapter. The **LookUp** window should look similar to Fig. 1.



LookUp identifies sequence database entries by name, accession number, author, organism, keyword, title, reference, feature, definition, length, or date. The output is a list of sequences.

LookUp of: Selected sequences from Main List.

○ Search the sequences chosen in the Main Window
● Search the chosen sequence libraries

☐ GenBank ☐ EMBL ☒ SwissProt ☐ NRL_3D
☐ GenBank EST ☐ EMBL EST ☒ SPTREMBL
☐ GenBank GSS ☐ EMBL GSS ☐ PIR

All text:
Accession:
Definition: ion & factor & alpha
Organism: azoa | viridiplant)
Author:
Reference:
Keyword:
Pub. title:

Command Line:

```
LookUp -LIBrary=SWISSPROT,SPTREMBL -DEFInition="elongation & factor" -ORGanism="eukaryota ! ( fungi | metazoa
```

How: Background Job

Fig. 1. The GCG Package SeqLab LookUp window. LookUp is an SRS derivative that allows one to construct complex text-based sequence database queries. It produces GCG list file format output.

The program will display the results of the search. Scroll through the output and then **Close** the window. The beginning of the LookUp output file from the example follows in Table 1. Be careful that all of the proteins included in the output from any text searching program are appropriate. All may look correct here, but improper nomenclature and other database inconsistencies can cause problems. If you find inappropriate proteins upon reading the output, you can either edit the output file to remove them, or *CUT* them from the SeqLab Editor display after loading the list. If you use an editor, you can comment out the undesired sequences by placing an exclamation point, ! , in front of the unwanted lines.

Select the LookUp output file in the **SeqLab Output Manager.** This is a very important window and will contain all of the output from your current SeqLab session. Files may be displayed, printed, saved in other locations with other names, and deleted from this window. Press the **Add to Main List** button in the **SeqLab Output Manager** and **Close** the window afterwards. Next, be sure that the LookUp output file is selected in the **SeqLab Main Window** and then switch the **Mode:** to **Editor.** This will load the file into the SeqLab Editor to permit further analyses. Notice that all of the sequences now appear in the Editor window with the amino acid residues color-coded. The nine color groups are based on a UPGMA clustering of the BLOSUM62 amino acid scoring matrix, and approximate physical property categories for the different amino acids. Expand the window to an appropriate size by *grabbing* the bottom-left corner of its *frame* and *pulling* it out as far as desired. Use the vertical scroll bar to

Table 1
LookUp Output File[a]

```
!!SEQUENCE_LIST 1.0
LOOKUP in: swissprot,sptrembl of: “([SQ-DEF: elongation* & factor* & alpha*] &
[SQ-ORG: eukaryota* ! ( fungi* | metazoa* | viridiplant* )])”
 79 entries May 10, 2001 16:08 ..
SWISSPROT:EF11_EUPCR ! ID: ef470001
! DE ELONGATION FACTOR 1-ALPHA 1 (EF-1-ALPHA-1).
! GN EFA1.
SWISSPROT:EF12_EUPCR ! ID: f8470001
! DE ELONGATION FACTOR 1-ALPHA 2 (EF-1-ALPHA-2).
! GN EFA2.
SWISSPROT:EF1A_BLAHO ! ID: 0d480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA).
SWISSPROT:EF1A_CRYPV ! ID: 14480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA).
SWISSPROT:EF1A_DICDI ! ID: 16480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA) (50 KDA ACTIN-BINDING PROTEIN)
! DE (ABP-50).
! GN EFAA.
SWISSPROT:EF1A_EIMBO ! ID: 17480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA) (FRAGMENT).
SWISSPROT:EF1A_ENTHI ! ID: 18480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA).
SWISSPROT:EF1A_EUGGR ! ID: 19480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA).
! GN TEF.
SWISSPROT:EF1A_GIALA ! ID: 1a480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA) (14 NM FILAMENT-ASSOCIATED
! DE PROTEIN) (FRAGMENT).
! GN TEF1.
SWISSPROT:EF1A_PLAFK ! ID: 2a480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA).
! GN MEF-1.
SWISSPROT:EF1A_STYLE ! ID: 35480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA).
! GN EFAA.
SWISSPROT:EF1A_TETPY ! ID: 38480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA) (14 NM FILAMENT-ASSOCIATED
! DE PROTEIN).
SWISSPROT:EF1A_TRYBB ! ID: 3d480001
! DE ELONGATION FACTOR 1-ALPHA (EF-1-ALPHA).
! GN TEF1.
SWISSPROT:EF1C_PORPU ! ID: 54480001
! DE ELONGATION FACTOR 1-ALPHA C (EF-1-ALPHA).
! GN TEF-C.
/////////////////////////////////////////////////////////////////////////
```

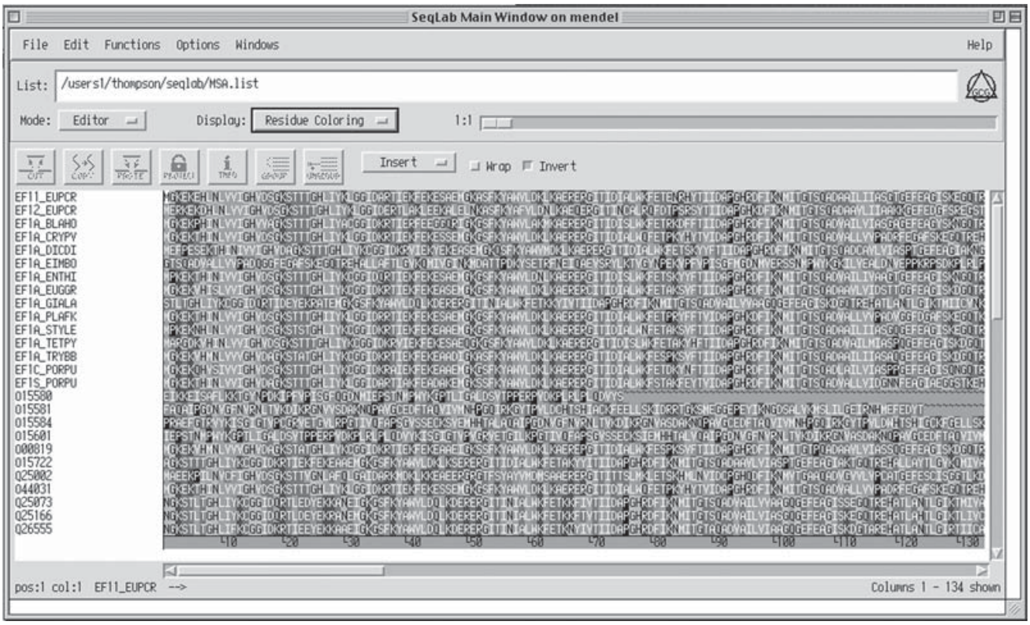
[a] Abridged screen trace of GCG’s LookUp output file. Notice the *list file* format that can be read by Wisconsin Package interfaces and programs, such as SeqLab and PileUp.

See companion CD for color Fig. 2

view the entire collection. Any portion of, or the entire alignment loaded, is now available for analysis by any of the GCG programs (*see* Fig. 2).

Alternatively, one can import sequences into SeqLab using the **Add sequences from Sequence Files. . .** choice under the **File** menu. Only GCG format compatible sequences or list files are accessible through this route. Using SeqLab’s Editor **File** menu **Import** function, one can directly load GenBank sequences or ABI binary trace files. You can also directly load sequences from the online GCG databases with the **Databases. . .** choice under the **Add sequences** menu if you know their proper identifier name or accession code. The **Add Sequences** window’s **Filter** box is very important! By default files are filtered such that only those that end with the extension *.seq* are displayed. To view all files, delete the **.seq** extension in the **Filter** box (including the period); be sure to leave the * wild card. Press the **Filter** button to display all of the files in your working directory. Select the file that you want from the **Files** box, and then check the **Add** and then **Close** buttons on the bottom of the window to put the desired file into your current list, if you’re in List Mode, or directly into the Editor, if you’re in **Editor Mode.**

Each protein sequence is listed by its official SwissProt or SPTREMBL entry name (ID identifier). The scroll bar at the bottom allows you to move through the sequences linearly. The side scroll bar allows you to scroll through all of your entries vertically.



SeqLab Main Window on mendel

List: /users1/thompson/seqlab/MSA.list

Mode: Editor

Display: Residue Coloring

1:1

Insert

□ Wrap ☒ Invert

pos:1 col:1 EF11_EUPCR -->

Columns 1 - 134 shown

Fig. 2. The SeqLab Editor window with a LookUp dataset loaded and ready to analyze.

Double-clicking on the various entry names or a single click on the **INFO** icon with the sequence entry name selected will provide the database and documentation. This is the same information that you can get with the GCG command **typedata -ref** at the command line. You can also change the sequences' names and add any documentation that you want in this window. Change the **Display:** box from **Residue Coloring** to **Feature Coloring** and then to **Graphic Features.** The display will show a schematic of the feature information from each entry with colors based on the information from the database Feature Table for the entry. **Graphic Features** present features using the same colors but in a *cartoon* fashion. Double-click on one of the various colored regions of the sequences (or use the **Features** choice under the **Windows** menu) to produce a new window that describes the features located at the cursor. Selecting the feature will show the detailed entry. All the features are fully editable through the **Edit** check box in this panel and new features can be added with desired shapes and colors through the **Add** check box.

Nearly all GCG programs are accessible through the **Functions** menu. To perform different analyses, select the entry of interest and then go to the **Functions** menu. You can select sequences in their entirety by clicking on their names or you can select any position(s) within sequences by *capturing* them with the mouse. You can select a range of sequence names by <shift><clicking> the top-most and bottom-most name desired, or <ctrl><click> sequence entry names to select noncontiguous entries. The **pos:** and **col:** indicators show you where the cursor is located on a sequence without including gaps and with including gaps respectively. The **1:1** scroll bar near the upper right-hand corner allows you to **zoom** in or out on the sequences; move it to 2:1 and beyond and notice the difference in the display.

It is a good idea to save the sequences in the display at this point and multiple times as you work on an alignment. Do this occasionally while you are in SeqLab just in case there is an interruption of service. Go to the **File** menu and choose **Save As.** Accept the default *.rsf* extension but give it any file name and directory specification you choose. Rich Sequence Format (RSF) contains all the aligned sequence data as well as all the reference and feature annotations associated with each entry. It is *Richer* than many other multiple sequence formats and is SeqLab's default format.

Similarity Searching to Increase (or Decrease) Dataset Size

A logical next step to preparing a multiple sequence alignment can be a similarity search to add similar sequences from the database to the dataset. Similarity searching can be carried out within SeqLab several ways. The BLAST and FastA family of programs are some of the more frequently used programs. An advantage to running any similarity search within the context of GCG is the results are immediately available for further analyses without the need for any sequence downloading or reformatting. This is achieved by virtue of the GCG list file format and the fact that all of the databases are mounted locally.

Database similarity searching can also be helpful by allowing one to sort a collection of sequences in order of alignment significance. This allows one to remove undesired sequences from the bottom of a list. But, be warned, on some systems you cannot run FastA in GCG on too small of a dataset without causing core dumps! To avoid this problem you can add another small database such as NRL_3D to your Search List Set as well as the list file that you would like to sort. This provides the necessary background randomization for normalization. Another point to remember, BLAST programs cannot be used to search against any sequence set that has not been preformatted into a BLAST compatible database.

In this example, the *Giardia* sequence will be used as a search query because many researchers consider *Giardia*'s most ancient ancestor to be rooted near the base of the universal tree of life on the eukaryote lineage. This is appropriate when examining *lower* eukaryotes. Select the **EF1A_GIALA** sequence entry name in the Editor display and launch FastA from the **Functions Database Sequence Searching** menu. If a **Which selection** window pops up asking if you want to use the **selected sequences** or **selected region,** choose **selected sequences** to execute the program on the full length of the selected protein. At most sites, the default protein database to search, **Search Set. . .** will be **Using pir:***. To specify the my LookUp output list file instead, push the **Search Set. . .** button, select **pir:*** in the **Build FastA's Search Set** box that pops up and then **Remove from Search Set.** Next, press the **Add Main List Selection. . .** button and then select your previous LookUp output list file from the **List Chooser** window that pops up, press **Add to Search Set.** Repeat this process using the **Add Database Sequences. . .** button and specify **NRL_3D** as discussed earlier. **Close** the **List Chooser** and the **Build Search Set** windows. The other parameters in the main **FastA** window are usually fine at their default settings, though you may want to decrease their cutoff Expectation-value to reduce the output list size. Press the **Options. . .** button, some of which can be helpful. Scroll down the window and notice the **Show sequence alignments in the output file** button. This toggles the command line option **-NoAlign** off and on to suppress the pairwise alignment section. This can be helpful if you are not interested in the pairwise alignments and wish to produce smaller output files. Restricting your search by the database sequence length or by date of their deposition in the database can also be useful. **Close** the **Options** window, be sure that the **FastA** program window shows **How: Background Job,** and then press the **Run** button. To check on the progress of the job you can go to SeqLab's **Windows** menu and choose **Job Manager.** Select the **FastA** entry to see its progress and then close the window.

The output is a GCG list file that can serve as input to other GCG programs such as PileUp, complete with beginning and ending attributes and complementary strand attributes when necessary, if using DNA. As in BLAST reports, the Expectation function, *E()*, is the most important statistic on the output. It is the likelihood (Expectations) that the match was observed by chance. Similar to BLAST *E-values*, the value in parenthesis describes the number of search set sequences that would be needed to obtain a *z-score* greater than or equal to the *z-score* obtained in any search purely by chance. The entries are sorted by this *z-score* parameter based on a normalization of the opt scores and their distribution from the rest of the database. However, this *z-score* is different from a Monte Carlo-style Gaussian distribution Z score. It is calculated as a simple linear regression against the natural log of the overall search set sequence length. Just like BLAST *E-values*, the smaller the number, the better and more significant the match. As a conservative rule-of-thumb, for a search against a random protein database of approx 10,000 sequences, as long as optimization is not turned off, *E()* scores much less than 0.01 are probably homologous, and scores from 0.01 to 1 may be homologous, whereas scores between 1 to 10 are only perhaps homologous, and scores above 10 are most probably not homologous. These guidelines can be skewed by compositional biases of the query and/or of the database. You should observe a demarcation where the scores drop off between the significant hits and background noise. Many significant E values can be seen in the output with scores of 10^{-100} or less, then there is a gradual increase in values from 10^{-99} through 10^{-75}, and finally many scores not quite as significant are seen with E values of 10^{-40} and above. A histogram of the score distribution is also displayed in the FastA outputs. This can be helpful to understand the statistical significance of the search and to ascertain whether the search list was of sufficient size. For the search statistics to be valid, the expected distribution, as indicated by the line of asterisks, should approximate the actual distribution, as shown by the equal signs. You want your list size big enough to include some random low scores to ascertain the significance of the alignments. The default FastA Expectation cutoff of 10.0 assures this. Table 2 shows a blowup of the highly significant score end of the graph—these are the best alignments found by the program, not the worst! The histogram can be suppressed with the **-NoHistogram** option if desired. FastA output also shows a sequence alignment for each pair up to a set number, unless you suppress this with the **-NoAlign** option, as in Table 2.

Use the **Output Manager** to load the dataset into the SeqLab Editor. Select the FastA output file in the **Output Manager** window and then press the **Add to Editor** button. Specify **Overwrite old with new** in the **Reloading Same Sequence** window when prompted, to take the FastA output and merge it with the sequences already in the open Editor. Click **Interrupt Loading** in the **Loading sequences** window after thirty or so sequences to prevent the entire file from loading. They are loaded in order of the FastA file and are, therefore, in order of similarity to your query. You are asked whether to **Modify the sequences** or **Ignore all attributes** in the **List file attributes set** window. The answer will depend on the type of alignment you are creating and the biological questions that you are asking. In many cases, especially if you are

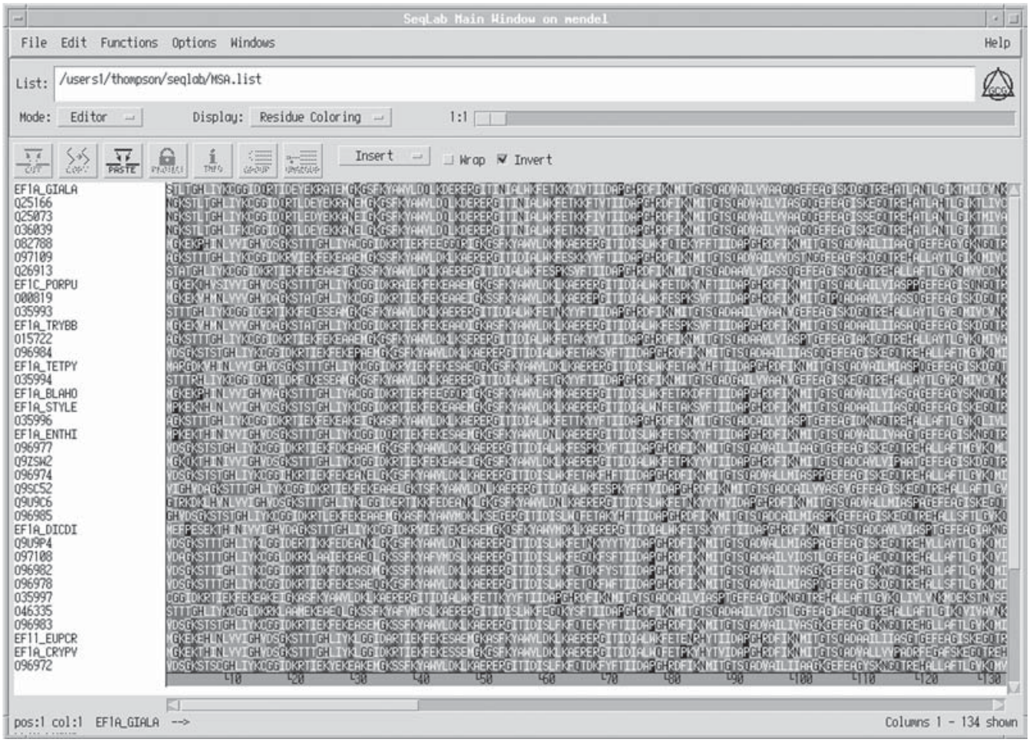
Table 2
FastA Output List File[a]

```
!!SEQUENCE_LIST 1.0
(Peptide) FASTA of: input_25.rsf{ef1a_giala} from: 1 to: 396 May 14, 2001 12:43
Description: Q08046 Giardia lamblia (Giardia intestinalis). elongation factor
 1-alpha (ef-1-a Accession/ID: Q08046
===================General comments===================
ID EF1A_GIALA STANDARD; PRT; 396 AA.
 TO: @/users1/thompson/.seqlab-mendel/fasta_ss1_25.list Sequences: 23,370
Symbols: 4,555,867 Word Size: 2
 Databases searched:
 SWISS-PROT, Release 39.0, Released on 15Jun2000, Formatted on 18Sep2000
 SPTREMBL, Release 14.0, Released on 15Jun2000, Formatted on 20Sep2000
 NRL_3D, Release 27.0, Released on 30Mar2000, Formatted on 2Oct2000
 Scoring matrix: GenRunData:blosum50.cmp
 Variable pamfactor used
 Gap creation penalty: 12 Gap extension penalty: 2
Histogram Key:
 Each histogram symbol represents 39 search set sequences
 Each inset symbol represents 3 search set sequences
 z-scores computed from opt scores
z-score obs exp
 (=) (*)
 20 745 0:====================
 22 0 0:
 24 0 0:
 26 0 0:
 28 1 5:*
 30 6 31:*
 32 92 120:===*
 34 371 325:========*=
 36 571 668:=============== *
 38 1711 1104:============================*===============
 40 1537 1540:=======================================*
 42 1997 1883:================================================*===
 44 1775 2077:============================================== *
 46 1782 2116:============================================== *
 48 2290 2025:===================================================*=======
 50 1475 1848:====================================== *
 52 1702 1625:=========================================*==
 54 1490 1388:===================================*===
 56 948 1159:========================= *
 58 898 952:========================*
 60 935 771:===================*====
 62 708 618:===============*===
 64 365 492:========== *
 66 402 389:=========*=
 68 419 306:=======*===
 70 343 240:======*==
 72 187 187:====*
 74 98 146:===*
 76 87 114:==*
 78 76 88:==*
 80 63 69:=*
 82 42 52:=*
 84 28 42:=*
 86 23 32:*
 88 25 25:*
 90 4 19:*
 92 0 15:* : *
 94 12 12:* :===*
 96 30 9:* :==*=======
 98 12 7:* :==*=
 100 5 5:* :=*
 102 4 4:* :=*
 104 1 3:* :*
 106 4 2:* :*=
 108 3 2:* :*
 110 0 1:* :*
 112 0 1:* :*
 114 0 1:* :*
 116 0 1:* :*
 118 0 1:* :*
>120 103 0:=== *==================================
Joining threshold: 37, opt. threshold: 25, opt. width: 16, reg.-scaled
The best scores are: init1 initn opt z-sc E(22522)..
SWISSPROT:EF1A_GIALA Begin: 1 End: 396
! Q08046 Giardia lamblia (Giardia int... 2696 2696 2696 3151.0 2.4e-169
SP_INVERTEBRATE:Q25166 Begin: 4 End: 399
! Q25166 diplomonad atcc50330. elonga... 2318 2318 2318 2709.5 9.3e-145
SP_INVERTEBRATE:Q25073 Begin: 4 End: 399
! Q25073 hexamita inflata. elongation... 2125 2125 2125 2484.1 3.3e-132
SP_INVERTEBRATE:O36039 Begin: 4 End: 401
! O36039 spironucleus vortens. elonga... 1071 2074 2083 2435.0 1.8e-129
SP_PLANT:O82788 Begin: 21 End: 417
! O82788 blastocystis hominis. elonga... 1017 1825 1830 2138.9 5.6e-113
SP_INVERTEBRATE:O97109 Begin: 4 End: 405
! O97109 naegleria andersoni. elongat... 802 1803 1827 2135.9 8.2e-113
SP_INVERTEBRATE:Q26913 Begin: 1 End: 395
! Q26913 trypanosoma cruzi. elongatio... 1003 1791 1815 2122.1 4.8e-112
SWISSPROT:EF1C_PORPU Begin: 21 End: 419
! P50256 porphyra purpurea. elongatio... 952 1786 1801 2104.8 4.4e-111
//////////////////////////////////////////////////////////////////////////
```

Table 2 *(continued)*
FastA Output List File[a]

```
SP_INVERTEBRATE:O96975 Begin: 6 End: 398
! O96975 euplotes aediculatus. transl... 935 1618 1693 1979.4 4.3e-104
SP_INVERTEBRATE:O77447 Begin: 21 End: 413
! O77447 plasmodium knowlesi. elongat... 928 1637 1684 1968.3 1.8e-103
SWISSPROT:EF1A_PLAFK Begin: 21 End: 413
! Q00080 plasmodium falciparum (isola... 922 1624 1671 1953.1 1.3e-102
SP_INVERTEBRATE:O77478 Begin: 21 End: 413
! O77478 plasmodium berghei. elongati... 915 1615 1662 1942.6 4.8e-102
SP_INVERTEBRATE:O44031 Begin: 21 End: 397
! O44031 cryptosporidium parvum. elon... 938 1453 1650 1929.2 2.7e-101
SP_INVERTEBRATE:O96976 Begin: 6 End: 398
! O96976 euplotes aediculatus. transl... 850 1578 1635 1911.7 2.5e-100
SP_INVERTEBRATE:Q9UAF6 Begin: 1 End: 346
! Q9uaf6 pyrsonympha grandis. elongat... 959 1518 1520 1778.5 6.6e-93
SP_INVERTEBRATE:Q9UAF5 Begin: 1 End: 346
! Q9uaf5 pyrsonympha grandis. elongat... 959 1515 1517 1775.0 1e-92
SP_INVERTEBRATE:Q9Y1W1 Begin: 1 End: 346
! Q9y1w1 pyrsonympha grandis. elongat... 955 1497 1499 1754.0 1.5e-91
SWISSPROT:EF12_EUPCR Begin: 22 End: 412
! Q27140 euplotes crassus. elongation... 581 1398 1483 1733.6 2.1e-90
//////////////////////////////////////////////////////////////////////////
SP_PLANT:O82555 Begin: 1 End: 290
! O82555 blastocystis hominis. elonga... 894 1233 1244 1457.5 5.1e-75
SP_INVERTEBRATE:O36034 Begin: 1 End: 178
! O36034 hexamita inflata. elongation... 901 901 901 1059.7 7.2e-53
SP_INVERTEBRATE:O36038 Begin: 1 End: 179
! O36038 spironucleus muris. elongati... 466 754 785 925.0 2.3e-45
SP_INVERTEBRATE:Q94839 Begin: 1 End: 406
! Q94839 glugea plecoglossi. elongati... 386 869 780 912.5 1.1e-44
SP_INVERTEBRATE:Q25002 Begin: 21 End: 426
! Q25002 glugea plecoglossi. peptide ... 386 869 780 912.2 1.2e-44
SP_INVERTEBRATE:O15600 Begin: 18 End: 169
! O15600 entamoeba histolytica. elong... 730 756 773 911.1 1.4e-44
SP_INVERTEBRATE:O15601 Begin: 1 End: 157
! O15601 entamoeba histolytica. elong... 699 699 723 853.5 2.2e-41
SWISSPROT:EF1S_PORPU Begin: 21 End: 460
! P50257 porphyra purpurea. elongatio... 492 1281 666 778.3 3.4e-37
SP_INVERTEBRATE:O15584 Begin: 9 End: 166
! O15584 entamoeba histolytica. elong... 582 627 631 745.7 2.3e-35
NRL_3D:1EFT Begin: 50 End: 323
! translation elongation factor EF-Tu... 202 431 508 595.5 5.2e-27
NRL_3D:1TUIC Begin: 42 End: 315
! translation elongation factor EF-Tu... 202 431 507 594.5 5.9e-27
NRL_3D:1TUIA Begin: 42 End: 315
! translation elongation factor EF-Tu... 202 431 507 594.5 5.9e-27
NRL_3D:1TUIB Begin: 42 End: 315
! translation elongation factor EF-Tu... 202 431 507 594.5 5.9e-27
//////////////////////////////////////////////////////////////////////////
SP_INVERTEBRATE:O15581 Begin: 3 End: 107
! O15581 entamoeba histolytica. elong... 407 407 435 518.9 9.7e-23
SP_INVERTEBRATE:O15580 Begin: 1 End: 74
! O15580 entamoeba histolytica. elong... 239 239 281 342.7 6.3e-13
NRL_3D:1D2EC Begin: 29 End: 337
! elongation factor tu (ef-tu), chain... 187 449 255 300.2 1.5e-10
NRL_3D:1D2ED Begin: 29 End: 337
! elongation factor tu (ef-tu), chain... 187 449 255 300.2 1.5e-10
NRL_3D:1D2EB Begin: 29 End: 337
! elongation factor tu (ef-tu), chain... 187 449 255 300.2 1.5e-10
NRL_3D:1D2EA Begin: 29 End: 337
! elongation factor tu (ef-tu), chain... 187 449 255 300.2 1.5e-10
NRL_3D:1DG1H Begin: 41 End: 335
! elongation factor tu, chain H - bac... 168 369 240 282.9 1.3e-09
NRL_3D:1DG1G Begin: 41 End: 335
! elongation factor tu, chain G - bac... 168 369 240 282.9 1.3e-09
NRL_3D:1EFCA Begin: 42 End: 336
! elongation factor, chain A - bacteria 168 369 240 282.9 1.3e-09
NRL_3D:1EFCB Begin: 42 End: 336
! elongation factor, chain B - bacteria 168 369 240 282.9 1.3e-09
NRL_3D:1ETU2 Begin: 1 End: 79
! translation elongation factor EF-Tu... 153 208 208 252.9 6.4e-08
NRL_3D:1EFM2 Begin: 1 End: 78
! elongation factor Tu (trypsin-modif... 148 203 203 247.6 1.2e-07
NRL_3D:1EFUA2 Begin: 16 End: 280
! translation elongation factor EF-Tu... 126 327 192 228.0 1.5e-06
NRL_3D:1EFUC2 Begin: 16 End: 280
! translation elongation factor EF-Tu... 126 327 192 228.0 1.5e-06
\End of List
! Distributed over 1 thread.
! Start time: Mon May 14 12:43:00 2001
! Completion time: Mon May 14 12:43:39 2001
! CPU time used:
! Database scan: 0:00:11.4
! Post-scan processing: 0:00:03.0
! Total CPU time: 0:00:14.4
! Output File: /users1/thompson/seqlab/ef1a_giala_25.fasta
```

[a] An abridged output list file from GCG's implementation of FastA. A histogram of score distributions is plotted preceding the list portion of the file where *hits* are ranked statistically by Expection-value. Normally a pairwise alignment section would follow the list, but that was turned off in this run with the -NoAlign option.



List: /users1/thompson/seqlab/MSA.list

Mode: Editor

Display: Residue Coloring

1:1

pos:1 col:1 EF1A_GIALA -->

Columns 1 - 134 shown

Fig. 3. The SeqLab Editor loaded with sorted FastA output. FastA can be used as a tool to sort a list into ranked order based on similarity to a particular query. Any desired portion of this output can then be loaded into the SeqLab editor for further analysis.

See companion CD for color Fig. 3

asking phylogenetic questions, you will not want to modify the sequences. Load their full length to maximize the available signal. However, if you are using extremely diverse sequences and/or domains of sequences, then trimming the sequences to the most conserved portions identified by FastA can be very helpful. Figure 3 shows the Editor display after loading the top part of my FastA file. At this juncture go to the **File** menu and save the RSF file. **Overwrite** in the **File exists** box if you have used the same name for this file earlier. RSF files are quite large and there is no need to save all the versions of the data.

MEME

A powerful *de novo* motif discovery algorithm can be run before actually performing multiple sequence alignment. The Expectation Maximization algorithm uses Bayesian probabilities and unsupervised learning to identify conserved motifs among a group of unaligned, ungapped sequences. The motifs do not have to be in congruent order among the different sequences, i.e., it has the power to discover *unalignable* motifs between sequences. This characteristic differentiates MEME from most other profile building techniques. It is implemented as the MEME program and it produces output containing a multiple profile file as well as a readable report file. The profile output serves as input to MotifSearch.

Select all of the sequences in the Editor window to run MEME. Several methods are available for selecting multiple sequence entry names. Either drag the mouse through them all if they are all visible at once in the display, or **<shift><click>** on the top- and bottom-most entries (select nonadjacent entries with <ctrl><clicks>), or select **Select All** from the **Edit** menu. Launch **MEME** from the **Functions Multiple Comparisons** menu. A **Which selection** window may pop up, asking whether you want to use either the **selected sequences** or **selected region**; choose **selected sequences** to run the program on the entire set of sequences. The algorithm can be sped up at the cost of sensitivity by decreasing the number of motifs to be found, by restricting the number of motifs found to exactly one in each sequence, and/or by decreasing the allowable motif window size.

MEME output consists of two files; a **.meme** readable text file and a **.prf** multiple profile text file. MotifSearch will scan any dataset specified with the multiple profile file that MEME produced. Scanning the original *training* dataset will annotate those regions that MEME discovered in your SeqLab Editor RSF file. After alignment the MEME motifs that are alignable will all line up. Go to the **Database Sequence Searching** menu and select **MotifSearch. . .** Specify your **query profile(s),** the one you just made, and change the **Search set** to the RSF dataset that is loaded in the Editor. Be sure to activate **Save motif features to the RSF file.** The output will return a .rsf file on top. This file contains the SeqLab format feature data discovered by MEME in your dataset. The .ms file contains the readable results of the search in list file format with the Expectation-value statistics and the number of motif hits for each hit. After the list file portion a *Position diagram* schematically describes the hits in each sequence. This can be viewed by pressing the **Display** button in the Output Manager.

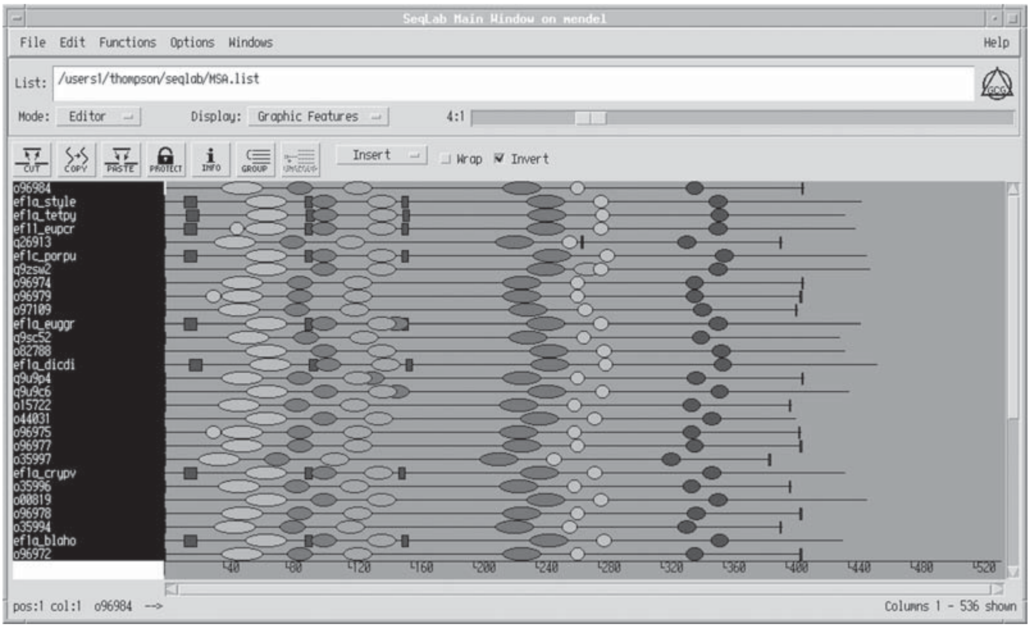
The Output Manager can be used to merge the motifsearch.rsf feature file with the existing data already open in the SeqLab Editor. This will add the feature annotation created with the MotifSearch-**RSF** option. The location of each motif will be included in the Editor sequence display. Use the **Add to Editor** Output Manager function. As noted earlier, specify **Overwrite old with new** in the next window when prompted. **Close** the **Output Manager** after loading the new RSF file. Change **Display:** to **Graphic Features** and check out the additional annotation. Figure 4 illustrates **Graphic Features** display at a **4:1** zoom ratio.

See companion CD for color Fig. 4

Searching PROSITE—GCG's Motifs

Many features have been described and catalogued in biological sequences over the years. Most have consensus patterns that allow one to screen an unknown sequence for their occurrence. One database of catalogued structural, regulatory, and enzymatic consensus patterns is Amos Bairoch's protein signature database, the *PROSITE Dictionary of Protein Sites and Patterns*. It is one of the quickest and easiest databases to search with a peptide sequence. The GCG program Motifs performs this search. The program can tolerate mismatches with a -MisMatch option and it displays an abstract with selected references for each motif signature found. In many cases this can be a tremendous aid that can suggest the function of an unknown peptide sequence. It can often lead to immediate answers and routes of investigation.

Start the Motifs program by selecting all of the protein entry names in SeqLab, then go to the **Functions Protein Analysis** menu and select **Motifs. . ..** The **Motifs** program window will be displayed. Check the **Save results as features in file motifs.rsf**



List: /users1/thompson/seqlab/MSA.list

Mode: Editor   Display: Graphic Features   4:1

pos:1 col:1 o96984 -->   Columns 1 - 536 shown

Fig. 4. Graphic Features display, MEME results. SeqLab can use *cartoons* to graphically display the feature annotation contained in sequence database entries and produced by programs such as MotifSearch. SeqLab merges this annotation with existing datasets with the **Add to Editor** and **Overwrite old with new** function. It also allows the user to *zoom in or out* on a dataset to see its entire length at once.

button in the **Motifs** program window. This file contains annotation discovered by the program. After running the program, the **motifs.rsf** file displayed. **Close** it and use the **Output Manager** to display the file with the **.motifs** extension. Notice the sites in the Motifs output file in Table 3 that have been characterized in these sequences and associated bibliography. This information can save a tremendous amount of work! Each site is shown with its sequence location below each consensus pattern. Post-translational modification sites found in many proteins, such as glycosylation, phosphorylation, amidation, and myristylation, will only be listed if the -Frequent option is specicifed. Realize that sites may be false positives. This is always a danger with simple consensus style searches. Notice that Motifs discovered the truly positive GTP-binding elongation factor signature and the ATP/GTP-binding P-loop site, yet it also found two probable false positives, the Prokaryotic membrane lipoprotein lipid attachment site and the FGGY family of carbohydrate kinases signature.

See companion CD for color Fig. 5

Close the **Motifs** output window, then load the motifs.rsf file into SeqLab. This will add the feature annotation created with the **RSF** option. The location of the PROSITE signatures will now be included in the Editor sequence display. Again use the **SeqLab Output Manager**. Select the file **motifs.rsf**, then press the **Add to Editor** button and specify **Overwrite old with new** to take the new motifs.rsf feature file and merge it with the old RSF file in the open Editor. **Close** the **Output Manager** after loading the RSF file. To display the new annotation, use **Features Coloring** or **Graphic Features**. Figure 5 shows the dataset using **Features Coloring** now annotated with its original database features as well as MEME discoveries and Motifs patterns.

Table 3
Abridged Motifs Output[a]

```
MOTIFS from: @/users1/thompson/.seqlab-mendel/motifs_54.list
Mismatches: 0 May 17, 2001 10:25 ..
input_54.rsf{GIARDIA_L} Check: 6084 Length: 475 ! In situ PileUp of: @/users
1/thompson/.seqlab-mendel/pileup_36.list
_______________________________________________________________________________________________________________
Efactor_Gtp D(K,R,S,T,G,A,N,Q,F,Y,W)x3E(K,R,A,Q)x(R,K,Q,D)(G,C)(I,V,M,K)(S,T)(I,V)x2(G,S,T,A,C,K,R,N,Q)
            D(Q)x{3} E(R)x(R)(G)(I)(T)(I)x{2} (A)
  64: YAWVL DQLKDERERGITINIA LWKFE
********************************************
* GTP-binding elongation factors signature *
********************************************
Elongation factors [1,2] are proteins catalyzing the elongation of peptide chains in protein biosynthesis. In both
prokaryotes and eukaryotes, there are three distinct types of elongation factors, as described in the following
table:
--------------------------------------------------------------------------------------------------------------
Eukaryotes  Prokaryotes  Function
--------------------------------------------------------------------------------------------------------------
EF-1alpha   EF-Tu        Binds GTP and an aminoacyl-tRNA; delivers the latter to the A site of ribosomes.
EF-1beta    EF-Ts        Interacts with EF-1a/EF-Tu to displace GDP and thus allows the regeneration of GTP-EF-1a.
EF-2        EF-G         Binds GTP and peptidyl-tRNA and translocates the latter from the A site to the P site.
--------------------------------------------------------------------------------------------------------------
The GTP-binding elongation factor family also includes the following proteins:
 - Eukaryotic peptide chain release factor GTP-binding subunits (3). These proteins interact with release factors
   that bind to ribosomes that have encountered a stop codon at their decoding site and help them to induce release
   of the nascent polypeptide. The yeast protein was known as SUP2 (and also as SUP35, SUF12 or GST1) and the human
   homolog as GST1-Hs.
 - Prokaryotic peptide chain release factor 3 (RF-3) (gene prfC). RF-3 is a class-II RF, a GTP-binding protein
   that interacts with class I RFs (see ERROR[Unused arguments - too many, or wrong type] in:<PDOC00607>) and
   enhance their activity (4).
 - Prokaryotic GTP-binding protein lepA and its homolog in yeast (gene GUF1) and in Caenorhabditis elegans
   (ZK1236.1).
 - Yeast HBS1 (5).
 - Rat statin S1 (6), a protein of unknown function which is highly similar to EF-1alpha.
 - Prokaryotic selenocysteine-specific elongation factor selB (7), which seems to replace EF-Tu for the insertion
   of selenocysteine directed by the UGA codon.
 - The tetracycline resistance proteins tetM/tetO (8,9) from various bacteria such as Campylobacter jejuni, En-
   terococcus faecalis, Streptococcus mutans and Ureaplasma urealyticum. Tetracycline binds to the prokaryotic
   ribosomal 30S subunit and inhibits binding of aminoacyl-tRNAs. These proteins abolish the inhibitory effect of
   tetracycline on protein synthesis.
 - Rhizobium nodulation protein nodQ (10).
 - Escherichia coli hypothetical protein yihK (11).
In EF-1-alpha, a specific region has been shown (12) to be involved in a conformational change mediated by the
hydrolysis of GTP to GDP. This region is conserved in both EF-1alpha/EF-Tu as well as EF-2/EF-G and thus seems typical
for GTP-dependent proteins which bind non-initiator tRNAs to the ribosome. The pattern we developed for this family
of proteins include that conserved region.
 -Consensus pattern: D-[KRSTGANQFYW]-x(3)-E-[KRAQ]-x-[RKQD]-[GC]-[IVMK]-[ST]- [IV]-x(2)-[GSTACKRNQ]
 -Sequences known to belong to this class detected by the pattern: ALL, except for 11 sequences.
 -Other sequence(s) detected in SWISS-PROT: NONE.
 -Last update: November 1997 / Text revised.
[ 1] Concise Encyclopedia Biochemistry (1988) 2nd ed., Walter de Gruyter, Berlin, NY.
[ 2] Moldave, K. (1985) Annu. Rev. Biochem. 54, 1109–1149.
[ 3] Stansfield, I., Jones, K. M., Kushnirov, V. V., et al. (1995) EMBO J. 14, 4365–4373.
[ 4] Grentzmann, G., Brechemier-Baey, D., Heurgue-Hamard, V., Buckingham, R. H. (1995) J. Biol. Chem. 270,
     10,595–10,600.
[ 5] Nelson, R. J., Ziegelhoffer, T., Nicolet, C., Werner-Washburne, M., Craig, E.A. (1992) Cell 71, 97–105.
[ 6] Ann, D. K., Moutsatsos, I. K., Nakamura, T., et al. (1991) J. Biol. Chem. 266, 10,429–10,437.
[ 7] Forchammer, K., Leinfeldr, W., Bock A. (1989) Nature 342, 453–456.
[ 8] Manavathu, E. K., Hiratsuka, K., Taylor, D. E. (1988) Gene 62, 17–26.
[ 9] Leblanc, D. J., Lee, L. N., Titmas, B. M., Smith, C. J., Tenover, F. C. (1988) J. Bacteriol. 170, 3618–3626.
[10] Cervantes, E., Sharma, S. B., Maillet, F., Vasse, J., Truchet, G., Rosenberg, C. (1989) Mol. Microbiol. 3,
     745–755.
[11] Plunkett, III, G., Burland, V. D., Daniels, D. L., Blattner, F. R. (1993) Nucleic Acids Res. 21, 3391–3398.
[12] Moller, W., Schipper, A., Amons, R. (1987) Biochimie 69, 983–989.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
///////////////////////////////////////////////////////////////////////////////////////////////////////////////
input_54.rsf{CRYPTOSPORIDIUM_P} Check: 6774 Length: 475 ! In situ PileUp of:
 @/users1/thompson/.seqlab-mendel/pileup_36.list
_______________________________________________________________________________________________________________
Atp_Gtp_A                (A,G)x4GK(S,T)
                          (G)x{4} GK(S)
              17: NLVVI     GHVDSGKS     TTTGH
****************************************
* ATP/GTP-binding site motif A (P-loop) *
****************************************
From sequence comparisons and crystallographic data analysis it has been shown [1,2,3,4,5,6] that an appreciable
proportion of proteins that bind ATP or GTPshare a number of more or less conserved sequence motifs. The best conserved
of these motifs is a glycine-rich region, which typically forms a flexible loop between a beta-strand and an alpha-
helix. This loop interacts with one of the phosphate groups of the nucleotide. This sequence motif is generally
referred to as the ‘A’ consensus sequence [1] or the ‘P-loop’ [5].
```

(continued)

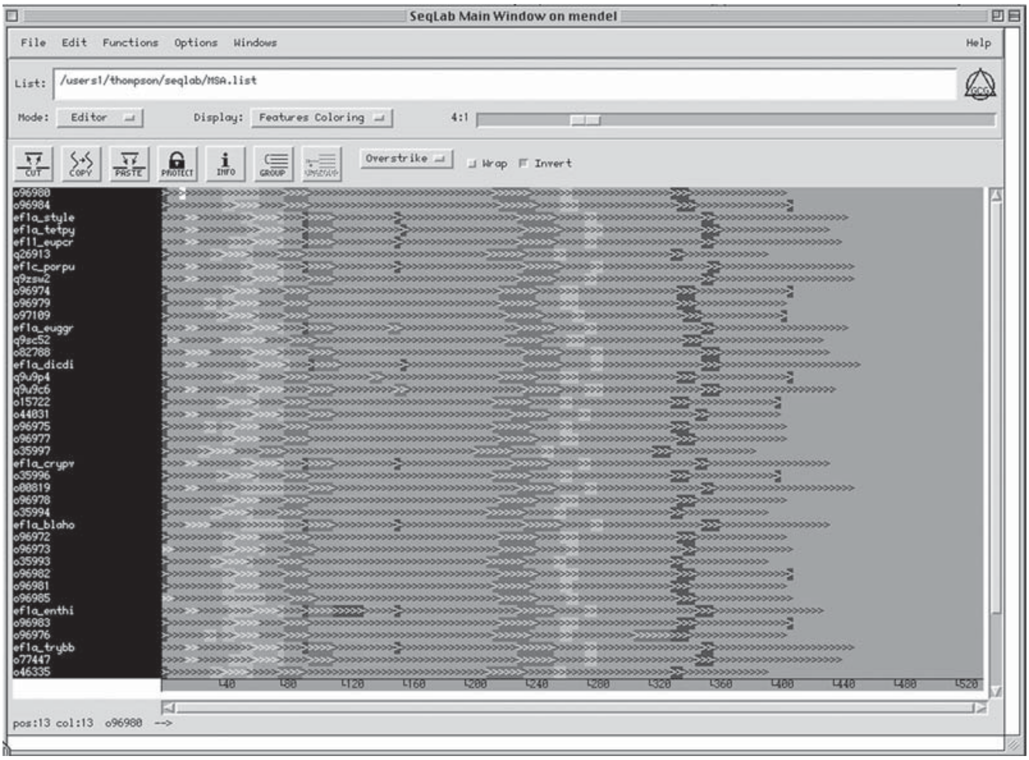
Table 3 *(continued)*
Abridged Motifs Output[a]

```
There are numerous ATP- or GTP-binding proteins in which the P-loop is found. We list below a number of protein
families for which the relevance of the presence of such motif has been noted:
 - ATP synthase alpha and beta subunits (see <PDOC00137>).
 - Myosin heavy chains.
 - Kinesin heavy chains and kinesin-like proteins (see <PDOC00343>).
 - Dynamins and dynamin-like proteins (see <PDOC00362>).
 - Guanylate kinase (see <PDOC00670>).
 - Thymidine kinase (see <PDOC00524>).
 - Thymidylate kinase (see <PDOC01034>).
 - Shikimate kinase (see <PDOC00868>).
 - Nitrogenase iron protein family (nifH/frxC) (see <PDOC00580>).
 - ATP-binding proteins involved in 'active transport' (ABC transporters) [7] (see <PDOC00185>).
 - DNA and RNA helicases [8,9,10].
 - GTP-binding elongation factors (EF-Tu, EF-1alpha, EF-G, EF-2, etc.).
 - Ras family of GTP-binding proteins (Ras, Rho, Rab, Ral, Ypt1, SEC4, etc.).
 - Nuclear protein ran (see <PDOC00859>).
 - ADP-ribosylation factors family (see <PDOC00781>).
 - Bacterial dnaA protein (see <PDOC00771>).
 - Bacterial recA protein (see <PDOC00131>).
 - Bacterial recF protein (see <PDOC00539>).
 - Guanine nucleotide-binding proteins alpha subunits (Gi, Gs, Gt, G0, etc.).
 - DNA mismatch repair proteins mutS family (See <PDOC00388>).
 - Bacterial type II secretion system protein E (see <PDOC00567>).
Not all ATP- or GTP-binding proteins are picked-up by this motif. A number of proteins escape detection because the
structure of their ATP-binding site is completely different from that of the P-loop. Examples of such proteins are the
E1-E2 ATPases or the glycolytic kinases. In other ATP- or GTP-binding proteins the flexible loop exists in a slightly
different form; this is the case for tubulins or protein kinases. A special mention must be reserved for adenylate
kinase, in which there is a single deviation from the P-loop pattern: in the last position Gly is found instead of Ser
or Thr.
- Consensus pattern: [AG]-x(4)-G-K-[ST]
- Sequences known to belong to this class detected by the pattern: a majority.
- Other sequence(s) detected in SWISS-PROT: in addition to the proteins listed above, the 'A' motif is also found
  in a number of other proteins. Most of these proteins probably bind a nucleotide, but others are definitively
  not ATP- or GTP-binding (as for example chymotrypsin, or human ferritin light chain).
- Expert(s) to contact by email: Koonin E.V.; koonin@ncbi.nlm.nih.gov
- Last update: July 1999 / Text revised.
[ 1] Walker, J. E., Saraste, M., Runswick, M. J., Gay, N. J. (1982) EMBO J. 1, 945–951.
[ 2] Moller, W., Amons, R. (1985) FEBS Lett. 186, 1–7.
[ 3] Fry, D. C., Kuby, S. A., Mildvan, A. S. (1986) Proc. Natl. Acad. Sci. USA 83, 907–911.
[ 4] Dever, T. E., Glynias, M. J., Merrick, W. C. (1987) Proc. Natl. Acad. Sci. USA 84, 1814–1818.
[ 5] Saraste, M., Sibbald, P. R., Wittinghofer, A. (1990) Trends Biochem. Sci. 15, 430–434.
[ 6] Koonin, E. V. (1993) J. Mol. Biol. 229, 1165–1174.
[ 7] Higgins, C. F., Hyde, S. C., Mimmack, M. M., Gileadi, U., Gill, D. R., Gallagher, M. P. (1990) J. Bioenerg.
     Biomembr. 22, 571–592.
[ 8] Hodgman, T. C. (1988) Nature 333, 22–23, Nature (Errata) 333, 578–578.
[ 9] Linder, P., Lasko, P., Ashburner, M., et al. (1989) Nature 337, 121–122.
[ 10] Gorbalenya, A. E., Koonin, E. V., Donchenko, A. P., Blinov, V. M. (1989) Nucleic Acids Res. 17, 4713–4730.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
input_54.rsf{BLASTOCYSTIS_H} Check: 172 Length: 475 ! In situ PileUp of: @/users1/thompson/.seqlab-mendel/
pileup_36.list
Atp_Gtp_A              (A,G)x4GK(S,T)
                        (G)x{4} GK(S)
            17: NLVVI     GHVVAGKS     TTTGH
Find reference above under sequence: input_54.rsf{CRYPTOSPORIDIUM_P} , pattern: Atp_Gtp_A.
______________________________________________________________________________________________________________________
Prokar_Lipoprotein ~(D,E,R,K)6(L,I,V,M,F,W,S,T,A,G)2(L,I,V,M,F,Y,S,T,A,G,C,Q) (A,G,S)C
                   ~(D,E,R,K){6} (L,I){2} (Y)(A)C
            24: VVAGK                      STTTGHLIYAC
         GGIDK
**********************************************************
* Prokaryotic membrane lipoprotein lipid attachment site *
**********************************************************
In prokaryotes, membrane lipoproteins are synthesized with a precursor signal peptide, which is cleaved by a specific
lipoprotein signal peptidase (signal peptidase II). The peptidase recognizes a conserved sequence and cuts upstream
of a cysteine residue to which a glyceride-fatty acid lipid is attached [1].
Some of the proteins known to undergo such processing currently include (for recent listings see [1,2,3]:
 - Major outer membrane lipoprotein (murein-lipoproteins) (gene lpp).
 - Escherichia coli lipoprotein-28 (gene nlpA).
 - Escherichia coli lipoprotein-34 (gene nlpB).
 - Escherichia coli lipoprotein nlpC.
 - Escherichia coli lipoprotein nlpD.
 - Escherichia coli osmotically inducible lipoprotein B (gene osmB).
 - Escherichia coli osmotically inducible lipoprotein E (gene osmE).
 - Escherichia coli peptidoglycan-associated lipoprotein (gene pal).
 - Escherichia coli rare lipoproteins A and B (genes rplA and rplB).
 - Escherichia coli copper homeostasis protein cutF (or nlpE).
 - Escherichia coli plasmids traT proteins.
 - Escherichia coli Col plasmids lysis proteins.
 - A number of Bacillus beta-lactamases.
```

Table 3 *(continued)*
Abridged Motifs Output[a]

```
 - Bacillus subtilis periplasmic oligopeptide-binding protein (gene oppA).
 - Borrelia burgdorferi outer surface proteins A and B (genes ospA and ospB).
 - Borrelia hermsii variable major protein 21 (gene vmp21) and 7 (gene vmp7).
 - Chlamydia trachomatis outer membrane protein 3 (gene omp3).
 - Fibrobacter succinogenes endoglucanase cel-3.
 - Haemophilus influenzae proteins Pal and Pcp.
 - Klebsiella pullulunase (gene pulA).
 - Klebsiella pullulunase secretion protein pulS.
 - Mycoplasma hyorhinis protein p37.
 - Mycoplasma hyorhinis variant surface antigens A, B, and C (genes vlpABC).
 - Neisseria outer membrane protein H.8.
 - Pseudomonas aeruginosa lipopeptide (gene lppL).
 - Pseudomonas solanacearum endoglucanase egl.
 - Rhodopseudomonas viridis reaction center cytochrome subunit (gene cytC).
 - Rickettsia 17 Kd antigen.
 - Shigella flexneri invasion plasmid proteins mxiJ and mxiM.
 - Streptococcus pneumoniae oligopeptide transport protein A (gene amiA).
 - Treponema pallidium 34 Kd antigen.
 - Treponema pallidium membrane protein A (gene tmpA).
 - Vibrio harveyi chitobiase (gene chb).
 - Yersinia virulence plasmid protein yscJ.
 - Halocyanin from Natrobacterium pharaonis (4), a membrane associated copper-binding protein. This is the first
   archaebacterial protein known to be modified in such a fashion).
From the precursor sequences of all these proteins, we derived a consensus pattern and a set of rules to identify this
type of post-translational modification.
-Consensus pattern: {DERK} (6)-[LIVMFWSTAG](2)-[LIVMFYSTAGCQ]-[AGS]-C [C is the lipid attachment site]
 Additional rules: 1) The cysteine must be between positions 15 and 35 of the sequence in consideration.
                   2) There must be at least one Lys or one Arg in the first seven positions of the sequence.
-Sequences known to belong to this class detected by the pattern: ALL.
-Other sequence(s) detected in SWISS-PROT: some 100 prokaryotic proteins. Some of them are not membrane
 lipoproteins, but at least half of them could be.
-Last update: November 1995 / Pattern and text revised.
[1] Hayashi, S., Wu, H. C. (1990) J. Bioenerg. Biomembr. 22, 451–471.
[2] Klein, P., Somorjai, R. L., Lau, P. C. K. (1988) Protein Eng. 2, 15–20.
[3] von Heijne, G. (1989) Protein Eng. 2, 531–534.
[4] Mattar, S., Scharf, B., Kent, S. B. H., Rodewald, K., Oesterhelt, D., Engelhard M. (1994) J. Biol. Chem. 269,
    14,939–14,945.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
input_54.rsf{PHYTOPHTHORA_I} Check: 9509 Length: 475 ! In situ PileUp of: @/users1/thompson/.seqlab-mendel/
pileup_36.list
Atp_Gtp_A            (A,G)x4GK(S,T)
                      (G)x{4} GK(S)
           17: ...VI    GHVDAGKS    TTTGH
Find reference above under sequence: input_54.rsf{CRYPTOSPORIDIUM_P} , pattern: Atp_Gtp_A.
Efactor_Gtp           D(K,R,S,T,G,A,N,Q,F,Y,W)x3E(K,R,A,Q)x(R,K,Q,D)(G,C)(I,V,M,K)(S,T)(I,V)x2(G,S,T,A,C,K,R,N,Q)
                      D(N)x{3} E(R)x(R)(G)(I)(T)(I)x{2} (A)
            64: YAWVL DNLKAERERGITIDIA LWKFE
Find reference above under sequence: input_54.rsf{GIARDIA_L} , pattern: Efactor_Gtp.
________________________________________________________________________________
Fggy_Kinases_1       (M,F,Y,G,S)x(P,S,T)x2K(L,I,V,M,F,Y,W)xW(L,I,V,M,F)x(D,E,N,Q,T,K,R)(E,N,Q,H)
                     (G)x(T)x{2} K(Y)xW(V)x(D)(N)
            53: EAAEL GKTSFKYAWVLDN LKAER
**************************************************
* FGGY family of carbohydrate kinases signatures *
**************************************************
It has been shown [1] that four different type of carbohydrate kinases seem tobe evolutionary related. These
enzymes are:
 - L-fucolokinase (EC 2.7.1.51) (gene fucK).
 - Gluconokinase (EC 2.7.1.12) (gene gntK).
 - Glycerokinase (EC 2.7.1.30) (gene glpK).
 - Xylulokinase (EC 2.7.1.17) (gene xylB).
 - L-xylulose kinase (EC 2.7.1.53) (gene lyxK).
These enzymes are proteins of from 480 to 520 amino acid residues. As consensus patterns for this family of kinases
we selected two conserved regions, one in the central section, the other in the C-terminal section.
-Consensus pattern: [MFYGS]-x-[PST]-x(2)-K-[LIVMFYW]-x-W-[LIVMF]-x-[DENQTKR]- [ENQH]
-Sequences known to belong to this class detected by the pattern: ALL, except for lyxK.
-Other sequence(s) detected in SWISS-PROT: 5.
-Consensus pattern: [GSA]-x-[LIVMFYW]-x-G-[LIVM]-x(7,8)-[HDENQ]-[LIVMF]-x(2)-[AS]-[STAIVM]-[LIVMFY]-[DEQ]
-Sequences known to belong to this class detected by the pattern: ALL.
-Other sequence(s) detected in SWISS-PROT: 11.
-Expert(s) to contact by email: Reizer J.; jreizer@ucsd.edu
-Last update: November 1997 / Patterns and text revised.
[1] Reizer, A., Deutscher, J., Saier, Jr., M. H., Reizer, J. (1991) Mol. Microbiol. 5, 1081–1089.
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
////////////////////////////////////////////////////////////////////////////////
```

[a] PROSITE patterns found by the GCG Motifs program in the example elongation factor dataset.



List: /users1/thompson/seqlab/MSA.list

Mode: Editor

Display: Features Coloring

4:1

pos:13 col:13 o96980 -->

Fig. 5. Motifs RSF annotation displayed by SeqLab. Motifs can create an RSF file with the location of PROSITE patterns annotated by color and shape. The display now shows annotation from the database, from the program pair MEME/MotifSearch, and from the program Motifs using **Features Coloring**.

Performing the Alignment Using the PileUp Program

To align all of these protein sequences, select all of the entries in the Editor window using one of the methods discussed. Once all of the sequences are selected, go to the **Functions** menu and select **Multiple comparison.** Click on **PileUp. . .** to align the entries. A new window will appear with the parameters for running PileUp. Be sure that the **How:** box indicates a **Background Job.** Often you'll accept all of the program defaults on a first run by pressing the **Run** button. As a default this would use the BLOSUM62 scoring matrix. However, the BLOSUM30 matrix can be helpful for aligning more divergent sequences. Depending on the level of divergence in a data set, better multiple sequence alignments can often be generated with alternate scoring matrices (the -Matrix option, specifies the desired matrix from the GCG logical directory GenMoreData) and/or different gap penalties. Beginning with GCG version 9.0, the BLOSUM62 matrix file, blosum62.cmp, is used as the default symbol comparison table. Furthermore, appropriate gap creation and extension penalties are now coded directly into the matrix, though they can still be adjusted within the program if desired. This is greatly improved over the normalized Dayhoff PAM 250 table and the program encoded penalty values formerly used. The BLOSUM series ranges from BLOSUM30 being appropriate for the most divergent datasets, to the BLOSUM100 table for the conserved datasets.

Click on the **Options** button. To specify the BLOSUM30 matrix select the check button next to and click on the **Scoring Matrix . . .** box in the **Pileup Options** window. This will launch a **Chooser for Scoring Matrix** window from which you can select the BLOSUM30 matrix file, **blosum30.cmp.** Double-click the matrix's name to see what it looks like; click **OK** to close both windows. Scroll through the rest of the **PileUp Options** window to see all the available options. **Close** it when finished and press **Run** in the **PileUp** window to launch the program.

The program first compares every sequence with every other one. This pairwise comparison will progressively merge the sequences into an alignment in order of similarity, from most to least similar. The window will disappear and then, depending on the complexity of the alignment and the load on the server, display a series of new output windows. The top window will be the Multiple Sequence Format (MSF) output from your PileUp run. Notice the BLOSUM30 matrix specification and the default gap introduction and extension penalties associated with that matrix, 15 and 5 respectively. As mentioned above, in most cases the default gap penalties will work fine with their respective matrices, though they can be changed if desired. As shown in the example in Table 4, notice the interleaved character of the sequences. They all have unique identities, addressable through their MSF filename together with their own name in braces, {name}.

The listing of sequence names near the top of the file contains the checksum. All GCG sequence programs use this number as a unique sequence identifier. There is a checksum line for the alignment as well as individual checksum lines for each member of the alignment. If any two of the checksum numbers are the same, then those sequences are identical. If they are, an editor can be used to place an exclamation point, ! at the start of the checksum line in the duplicate sequence. Exclamation points are interpreted by GCG as remark delineators, therefore, the duplicate sequence will be ignored in subsequent programs. Or the sequence could be **CUT** from the alignment with the SeqLab Editor. Similarly, the *Weight* designation determines how each sequence contributes to the alignment profile. Sometimes it is worthwhile to adjust these values so that the contribution of a collection of very similar sequences does not overwhelm the signal from the more divergent sequences.

Use the **Output Manager**, **Add to Editor**, and **Overwrite old with new** to take the new MSF output and merge it with the old RSF file in the open Editor. This will keep all of the database feature annotation intact, yet renumber all of the reference locations based on the gaps in the alignment. **Close** the **Output Manager** after loading the new alignment. The next window will contain PileUp's cluster dendrogram, in the EF-1α example, as shown in Fig. 6.

See companion CD for color Fig. 6

PileUp automatically creates this dendrogram of clustering similarity between the sequences. It can be very helpful for adjusting the sequence Weight values, to average the contribution of each sequence to a profile. The lengths of the vertical lines are proportional to the differences in similarity between the sequences. This tree is not an evolutionary tree, and it should never be presented as one. It is akin to an uncorrected Unweighted Pair Group Method with Arithmetic Mean (UPGMA) tree, prone to all the same UPGMA errors. If the rates of evolution for each lineage were exactly the same, then it could represent a *true* phylogenetic tree, but this is seldom the case in nature. No phylogenetic inference optimization criteria algorithm, such as maximum likelihood, least-squares fit, or parsimony, nor any molecular substitution, multiple hit correction models, such as Jukes-Cantor, Kimura, or any other subset of the

Table 4
GCG PileUp Output, Abridged[a]

```
!!AA_MULTIPLE_ALIGNMENT 1.0
PileUp of: @/users1/thompson/.seqlab-mendel/pileup_28.list
 Symbol comparison table: /usr/gcg/gcgcore/data/moredata/blosum30.cmp CompCheck: 8599
              GapWeight: 15
        GapLengthWeight: 5
 pileup_28.msf MSF: 472 Type: P May 14, 2001 14:35 Check: 2476 ..
 Name: ef1a_giala Len: 472 Check: 8631 Weight: 1.00
 Name:     q25166 Len: 472 Check: 6209 Weight: 1.00
 Name:     q25073 Len: 472 Check: 2914 Weight: 1.00
 Name:     o36039 Len: 472 Check: 7560 Weight: 1.00
 Name:     o96981 Len: 472 Check: 3858 Weight: 1.00
 Name:     o96980 Len: 472 Check: 3082 Weight: 1.00
 Name:     o44031 Len: 472 Check: 851 Weight: 1.00
 Name: ef1a_crypv Len: 472 Check: 2406 Weight: 1.00
 Name:     o77447 Len: 472 Check: 9210 Weight: 1.00
 Name:     o77478 Len: 472 Check: 1123 Weight: 1.00
 Name: ef1a_plafk Len: 472 Check: 1436 Weight: 1.00
//////////////////////////////////////////////////////////////
 Name:     o96978 Len: 472 Check: 6796 Weight: 1.00
 Name: ef1c_porpu Len: 472 Check: 6199 Weight: 1.00
 Name:     o46335 Len: 472 Check: 7668 Weight: 1.00
 Name:     o97108 Len: 472 Check: 5669 Weight: 1.00
 Name:     o97109 Len: 472 Check: 6457 Weight: 1.00
//
           1                                                   50
ef1a_giala ~~~~~~~~~~ ~~~~~~~~~~ ~~~STLTGHL IYKCGGIDQR TIDEYEKRAT
    q25166 ~~~~~~~~~~ ~~~~~~~~~~ NGKSTLTGHL IYKCGGIDQR TLDEYEKRAN
    q25073 ~~~~~~~~~~ ~~~~~~~~~~ NGKSTLTGHL IYKCGGIDQR TLEDYEKKAN
    o36039 ~~~~~~~~~~ ~~~~~~~~~~ NGKSTLTGHL IFKCGGIDQR TLDEYEKKAN
    o96981 ~~~~~~~~~~ ~~~~~~~~VD SGKSTSTGHL IYKCGGIDER TIEKFEKEAK
    o96980 ~~~~~~~~~~ ~~~~~~~~VD SGKSTSTGHL IYKCGGIDER TIEKFEKEAK
    o44031 ~~~MGKEKTH INLVVIGHVD SGKSTTTGHL IYKLGGIDKR TIEKFEKESS
ef1a_crypv ~~~MGKEKTH INLVVIGHVD SGKSTTTGHL IYKLGGIDKR TIEKFEKESS
    o77447 ~~~MGKEKTH INLVVIGHVD SGKSTTTGHI IYKLGGIDRR TIEKFEKESA
    o77478 ~~~MGKEKTH INLVVIGHVD SGKSTTTGHI IYKLGGIDRR TIEKFEKESA
ef1a_plafk ~~~MGKEKTH INLVVIGHVD SGKSTTTGHI IYKLGGIDRR TIEKFEKESA
    o96975 ~~~~~~~~~~ ~~~~~~~~VD SGKSTTTGHL IYKLGGTDAR TIEKFEKESA
    o96976 ~~~~~~~~~~ ~~~~~~~~VD SGKSTTTGHL IYKCGGIDAR TIEKFEKESA
ef11_eupcr ~~~MGKEKEH LNLVVIGHVD SGKSTTTGHL IYKLGGIDAR TIEKFEKESA
    o82788 ~~~MGKEKPH INLVVIGHVD SGKSTTTGHL IYACGGIDKR TIERFEEGGQ
ef1a_blaho ~~~MGKEKPH INLVVIGHVV AGKSTTTGHL IYACGGIDKR TIERFEEGGQ
    o96982 ~~~~~~~~~~ ~~~~~~~~VD SGKSTTIGHL IYKCGGIDKR TIDKFDKDAS
    o96983 ~~~~~~~~~~ ~~~~~~~~VD SGKSTSTGHL IYKCGGIDKR TIEKFEKEAS
    o96972 ~~~~~~~~~~ ~~~~~~~~VD SGKSTSCGHL IYKCGGIDKR TIEKYEKEAK
    o96973 ~~~~~~~~~~ ~~~~~~GHVD SGKSTSCGHL IYKCGGIDKR TIEKYEKEAN
ef1a_enthi ~~~MPKEKTH INIVVIGHVD SGKSTTTGHL IYKCGGIDQR TIEKFEKESA
    o35994 ~~~~~~~~~~ ~~~~~~~~~~ ~~~STTTRHL IYKCGGIDQR TLDRFQKESE
    o35993 ~~~~~~~~~~ ~~~~~~~~~~ ~~~STTTGHL IYKCGGIDER TIKKFEQESE
    q26913 ~~~~~~~~~~ ~~~~~~~~~~ ~~~STATGHL IYKCGGIDKR TIEKFEKEAA
    o00819 ~~~MGKEKVH MNLVVVGHVD AGKSTATGHL IYKCGGIDKR TIEKFEKEAA
ef1a_trybb ~~~MGKEKVH MNLVVVGHVD AGKSTATGHL IYKCGGIDKR TIEKFEKEAA
    o96977 ~~~~~~~~~~ ~~~~~~~~VD SGKSTSTGHL IYKCGGIDKR TIEKFDKEAA
ef1a_euggr ~~~MGKEKVH ISLVVIGHVD SGKSTTTGHL IYKCGGIDKR TIEKFEKEAS
    o35997 ~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~ ~~~CGGIDKR TIEKFEKEAK
    o35996 ~~~~~~~~~~ ~~~~~~~~~~ AGKSTTTGHL IYKCGGIDKR TIEKFEKEAK
ef1a_dicdi MEFPESEKTH INIVVIGHVD AGKSTTTGHL IYKCGGIDKR VIEKYEKEAS
    o15722 ~~~~~~~~~~ ~~~~~~~~~~ AGKSTTTGHL IYKCGGIDKR TIEKFEKEAA
    q9zsw2 ~~~MGKQKTH INIVVIGHVD SGKSTTTGHL IYKCGGIDKR TIEKFEKEAA
    q9sc52 ~~~~~~~~~~ ~~~~VIGHVD AGKSTTTGHL IYKCGGIDKR TIEKFEKEAA
    o96984 ~~~~~~~~~~ ~~~~~~~~VD SGKSTSTGHL IYKCGGIDKR TIEKFEKEPA
ef1a_style ~~~MPKEKNH LNLVVIGHVD SGKSTSTGHL IYKCGGIDKR TIEKFEKEAA
    o96979 ~~~~~~~~~~ ~~~~~~~~VD SGKSTTTGHL IYKCGGIDKR VIEKFEKESA
ef1a_tetpy ~~MARGDKVH INLVVIGHVD SGKSTTTGHL IYKCGGIDKR VIEKFEKESA
    o96985 ~~~~~~~~~~ ~~~~~~GHVD SGKSTSTGHL IYKCGGIDKR TLEKFEKEAA
    q9u9p4 ~~~~~~~~~~ ~~~~~~~~VD SGKSTTTGHL IYKLGGIDER TIKKFEDEAN
    q9u9c6 ~~GTRKDKLH VNLVVIGHVD SGKSTTTGHL IYKLGGIDER TIKKFEDEAN
    o96974 ~~~~~~~~~~ ~~~~~~~~VD SGKSTSTGHL IYKCGGIHKR TIEKFEKEAN
    o96978 ~~~~~~~~~~ ~~~~~~~~VD SGKSTTTGHL IYKCGGIDKR TIEKFEKESA
ef1c_porpu ~~~MGKEKQH VSIVVIGHVD SGKSTTTGHL IYKCGGIDKR AIEKFEKEAA
    o46335 ~~~~~~~~~~ ~~~~~~~~~~ ~~~STTTGHL IYKCGGLDKR KLAAMEKEAE
    o97108 ~~~~~~~~~~ ~~~~~~~~VD AGKSTTTGHL IYKCGGLDKR KLAAIEKEAE
    o97109 ~~~~~~~~~~ ~~~~~~~~~~ AGKSTTTGHL IYKCGGIDKR VIEKFEKEAA

           51                                                 100
ef1a_giala EMGKGSFKYA WVLDQLKDER ERGITINIAL WKFETKKYIV TIIDAPGHRD
    q25166 EMGKGSFKYA WVLDQLKDER ERGITINIAL WKFETKKFTV TIIDAPGHRD
    q25073 EIGKGSFKYA WVLDQLKDER ERGITINIAL WKFETKKFIV TIIDAPGHRD
    o36039 ELGKGSFKYA WVLDQLKDER ERGITINIAL WKFETKKFIV TIIDAPGHRD
    o96981 QIGKESFKYA WVLDKLKAER ERGITIDIAL WKFESQKYSF TIIDAPGHRD
    o96980 QIGKESFKYA GLLDILKAER ARGITIDIAL WKFESQKYSF TIIDAPGHRD
    o44031 EMGKGSFKYA WVLDKLKAER ERGITIDIAL WQFETPKYHY TVIDAPGHRD
ef1a_crypv EMGKGSFKYA WVLDKLKAER ERGITIDIAL WQFETPKYHY TVIDAPGHRD
    o77447 EMGKGSFKYA WVLDKLKAER ERGITIDIAL WKFETPRYFF TVIDAPGHKD
    o77478 EMGKGSFKYA WVLDKLKAER ERGITIDIAL WKFETPRYFF TVIDAPGHKH
ef1a_plafk EMGKGSFKYA WVLDKLKAER ERGITIDIAL WKFETPRYFF TVIDAPGHKD
    o96975 EMGKGTFKYA WVLDKLKAER ERGITIDIAL WKFETTNRFY TIIDAPGHRD
    o96976 EMGKGSFKYA FVLDNLKAER ERGITIDIAL WKFETPKRFY TIIDAPGHRD
ef11_eupcr EMGKASFKYA WVLDKLKAER ERGITIDIAL WKFETENRHY TIIDAPGHRD
    o82788 RIGKGSFKYA WVLDKMKAER ERGITIDISL WKFQTEKYFF TIIDAPGHRD
ef1a_blaho RIGKGSFKYA WVLAKMKAER ERGITIDISL WKFETRKDFF TIIDAPGHRD
///////////////////////////////////////////////////////////////////
```

[a] An abridged GCG PileUp output MSF file. The format holds the file name, type, date, and checksum, sequence names, checksums, lengths, and weights, and the aligned sequence data in an interleaved fashion.

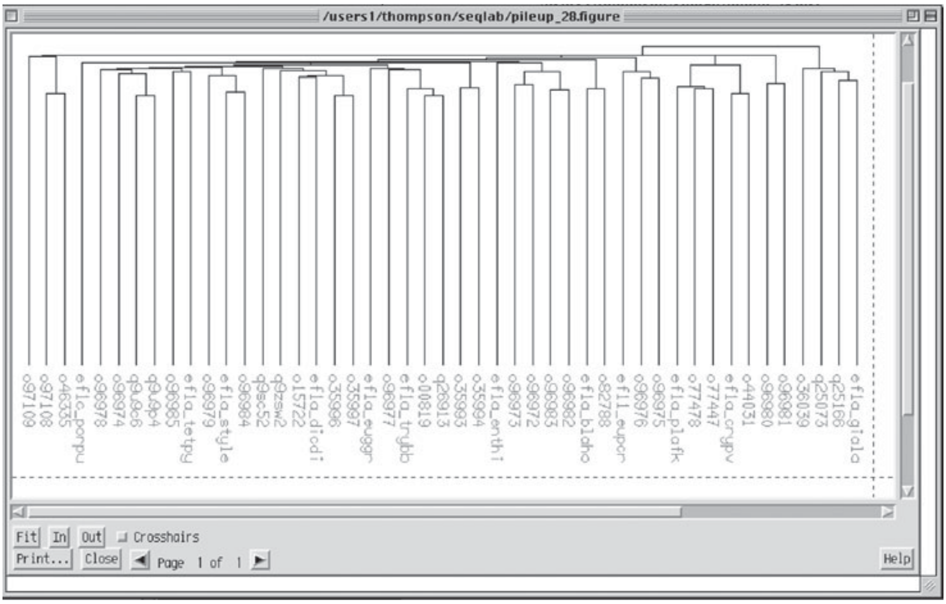


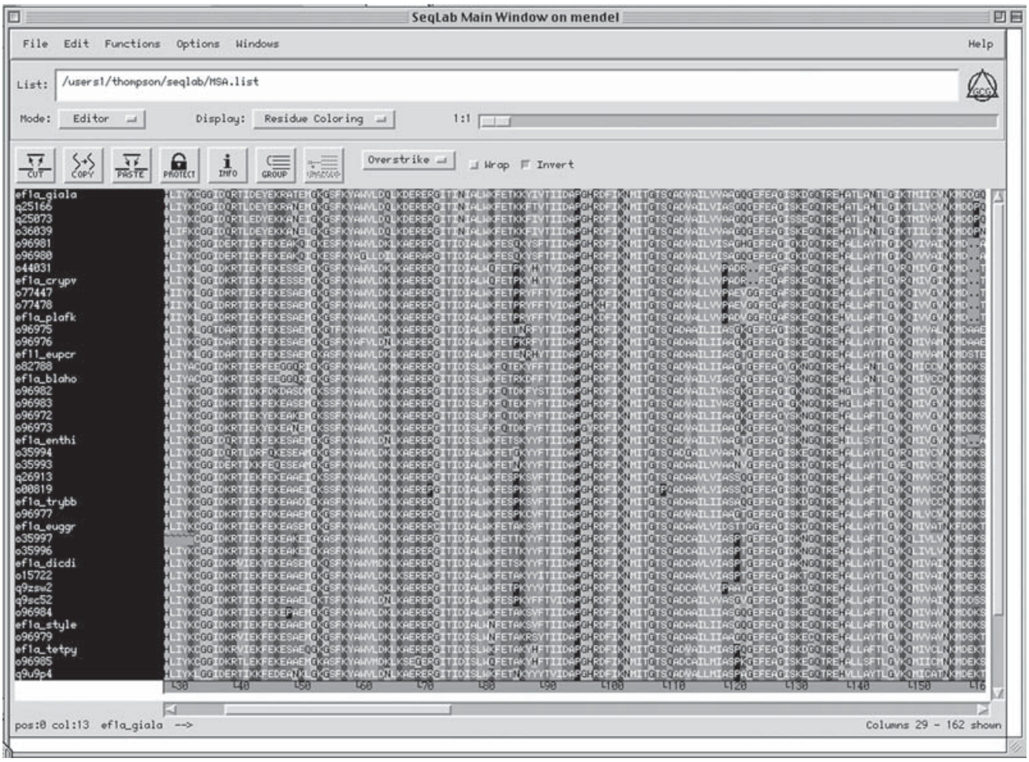
Fig. 6. PileUp's similarity dendrogram. The PileUp program automatically plots a cluster dendrogram of the similarities between the sequences of your dataset. The lengths of the vertical lines are proportional to those similarities.

General Time Reversible (GTR) model, nor any site rate heterogeneity models such as a Gamma correction, are used in its construction. PileUp's dendrogram merely indicates the relative similarity of the sequences based on the scoring matrix used and, is the clustering order used to create the alignment.

If desired, you can directly print from a SeqLab graphics Figure window to a PostScript file by selecting **Print . . .** Be sure that the **Output Device:** is **[Encapsulated] PostScript File.** You can rename the output file. Click **Proceed** to create an EPS file output in your current directory. To actually print this file you may need to transfer the file to a local machine attached to a PostScript printer unless you have direct access to the UNIX system printer and it is PostScript compatible. All Macintosh compatible laser printers use PostScript by default. Carefully check any laser printer connected to a Wintel system to be sure that it is PostScript compatible. **Close** the dendrogram window to return to the Editor.

See companion CD for color Figs. 7 and 8

After loading the MSF file using **Residue Coloring** and a **1:1** zoom ratio, the Editor display looks like Fig. 7 with the residues aligned by color. The columns of color represent columns of aligned residues. By changing the **Display:** box from **Residue Coloring** to **Graphic Features,** the display shows a schematic of the feature information from each entry, as well as all of the motifs discovered by the programs Motifs and MotifSearch, as in Fig. 8. Quick double-clicks on any of the color-coded feature regions in the Editor display will produce a **Features** window. Selecting the Feature entry in the new window reveals more information for that particular feature. Clicking once in the colored region and then using the **Features** option from the **Windows** menu will also produce the **Features** window. You should save this work as an updated RSF file!



List: /users1/thompson/seqlab/MSA.list

Mode: Editor   Display: Residue Coloring   1:1

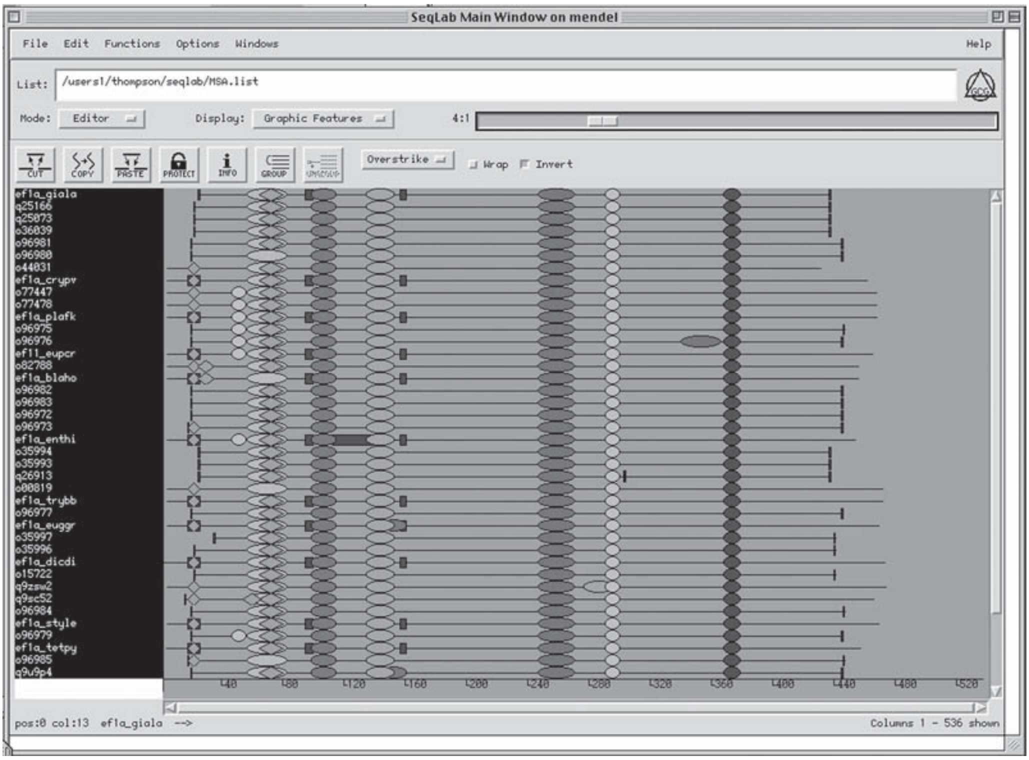
pos:0 col:13 ef1a_giala -->   Columns 29 - 162 shown

Fig. 7. Elongation Factor alignment displayed by the SeqLab Editor. The PileUp alignment loaded into the SeqLab Editor, displayed using **Residue Coloring** and a **1:1** zoom ratio.

Visualizing Conservation in Multiple Sequence Alignments

The most conserved portions of an alignment are those most resistant to evolutionary change, which often arise from structural constraint. The graphics program PlotSimilarity can be used to visualize positional conservation of a multiple sequence alignment. The program draws a graph of the running average similarity along with a group of aligned sequences (or of a profile with the **-Profile** option). The PlotSimilarity peaks of a protein alignment represent the areas that are most conserved and most resistant to evolutionary change. PlotSimilarity can also be helpful for ascertaining alignment quality by noting changes in the overall average alignment similarity and in those regions of conservation within the alignment, as it is adjusted and refined.

As before, select all of the sequence names and then go to the **Functions** menu and select **Multiple comparison PlotSimilarity . . . .** Check **Save SeqLab colormask to** and **Scale the plot between:** the **minimum and maximum values calculated from the alignment.** The first option's output file will be used in the next step. The second specification launches the program's command line **-Expand** option. This scales the plot, between the maximum and minimum similarity values observed, so that the entire graph is used, rather than just the portion of the Y axis that the alignment occupies. The Y-axis of the resulting plot uses the similarity values from the scoring matrix that was used to create the alignment unless you specify an alterna-



List: /users1/thompson/seqlab/MSA.list

Mode: Editor   Display: Graphic Features   4:1

Overstrike   □ Wrap   ☒ Invert

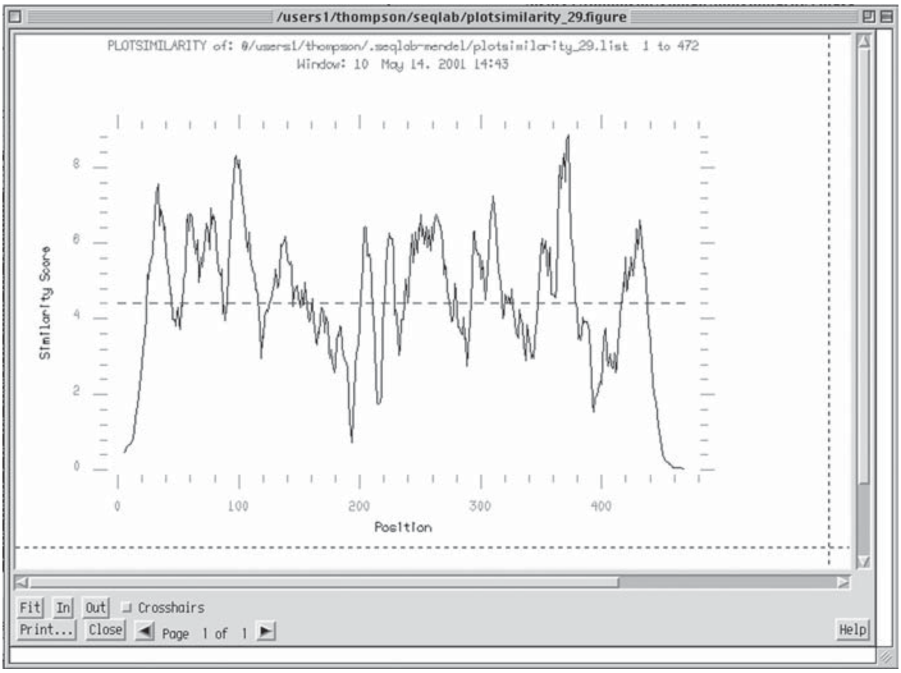
pos:0 col:13 ef1a_giala -->   Columns 1 - 536 shown

Fig. 8. The PileUp aligned dataset visualized with SeqLab's **Graphic Features** annotation and a **4:1** zoom ratio. Aligned annotation now includes database features, plus Motifs and MotifSearch patterns.

See companion CD for color Fig. 9

tive. The default matrix, BLOSUM62, begins its identity value at 4 and ranges up to 11 with mismatches as low as –4. **Close** the **Options** window; the **Command Line:** box now reflects updated options. Click the **Run** box to launch the program. The output will quickly return as shown in Fig. 9. **Close** the plotsimilarity.cmask display and the **Output Manager**.

Strong peaks of sequence similarity are seen centered around positions 30, 100, and 375. The dashed line across the middle shows the average similarity value for the entire alignment, approx 4.4. As before, to print a SeqLab graphics Figure to a PostScript file: select **Print . . .** from the Figure window, choose **Output Device: [Encapsulated] PostScript File,** and click **Proceed,** to create EPS file output. Take note where the similarity significantly falls off. In this example, the deepest valleys are the least similar regions of the alignment. They lay in the first 25 residues, a region around 190 and 220, around 390, and about the last 25 residues. **Close** the PlotSimilarity window. Go to the **File** menu and click on **Open Color Mask Files.** This will produce another window from which to select the new **plotsimilarity.cmask** file; click on **Add** and **Close** the window. This will produce a gray scale overlay on the sequences that describes their regional similarity where darker gray corresponds to higher similarity values as shown in Fig. 10. Notice the strong conservation peak near column 100 in the alignment, one of EF-1α's GTP binding regions.



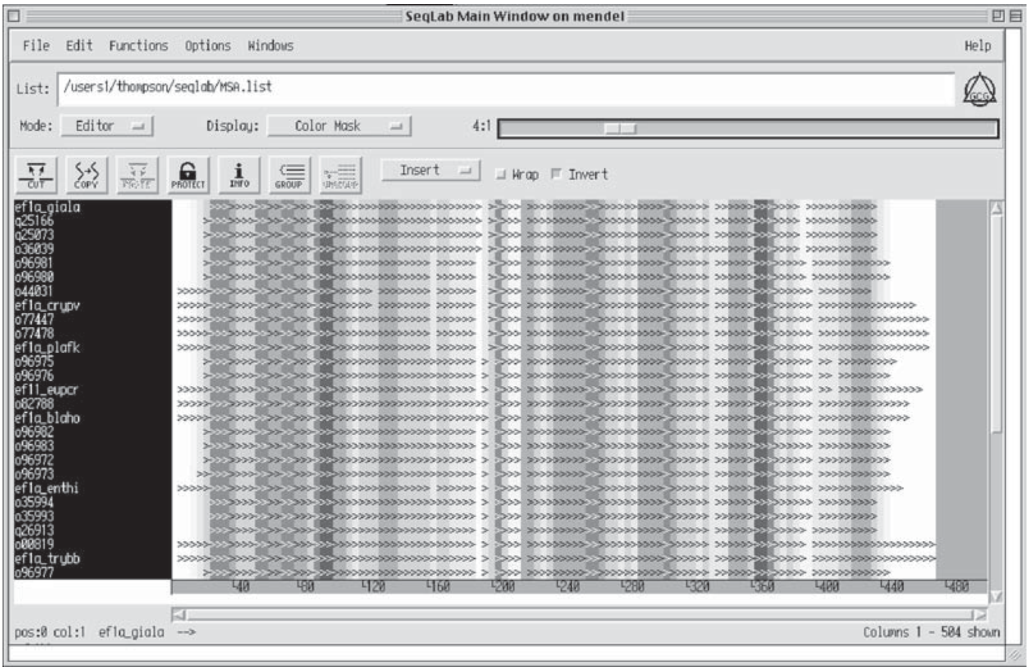
/users1/thompson/seqlab/plotsimilarity_29.figure

PLOTSIMILARITY of: @/users1/thompson/.seqlab-mendel/plotsimilarity_29.list 1 to 472
Window: 10 May 14. 2001 14:43

Fig. 9. The GCG PlotSimilarity graphic. PlotSimilarity draws a graph of the running similarity along the length of a multiple sequence alignment using a window averaging approach. Peaks are conserved regions; valleys are dissimilar areas.

Improving Alignments within SeqLab

You can now select those regions of low similarity to try to improve their alignment using the **-InSitu** option that realigns regions within an alignment. Be sure that all of your sequences are selected, then zoom back in your alignment to 1:1 so that you can see individual residues and scroll to the carboxy end. It is best to start at the carboxy termini so that the low similarity regions do not become skewed as you proceed. Select a region of low similarity across the complete sequence set. This can be accomplished using the mouse if it's all on the screen. Otherwise, use the **Edit Select Range** function. Determine the positions by placing your cursor at the beginning and end of the range to be selected and noting the column number in the lower left-hand of the Editor display. Once all of the sequences and the region that you wish to improve are selected, go to the **Functions** menu and again select **Multiple comparison.** Click on **PileUp . . .** to realign all of the sequences within that region. The **Windows** menu also contains a *shortcut* listing of all of the programs that have been accessed in the current session. You will be asked whether you want to use the **Selected sequences** or **Selected region**. It is very important to specify **Selected region.** This will produce a new window with the parameters for running PileUp. Next, be sure to click on **Options . . .** to change the way that PileUp will perform the alignment. In the **Options** window check the gap creation and extension boxes and change their respective values to much less than the default. Changing them to about a third of the default value works well. For the BLOSUM30 matrix change the values to 5 and 2. Most impor-



List: /users1/thompson/seqlab/MSA.list

Mode: Editor Display: Color Mask 4:1

pos:0 col:1 ef1a_giala --> Columns 1 - 504 shown

Fig. 10. PlotSimilarity Color Mask on an alignment. PlotSimilarity can produce a Color Mask that can be superimposed over an open alignment in the Editor. Dark regions now correspond to conserved peaks, whereas valleys are represented with white areas.

tantly, check **Realign a portion of an existing alignment**. This calls up the command line **-InSitu** option. Otherwise only that portion of the selected alignment will be retained in the output. Furthermore, deselect the **Plot dendrogram** box. **Close** the window and notice the new options in the PileUp Command Line: **Run** to improve your alignment. The window will disappear and your results will be returned quickly since you are only realigning a portion. The top window will be the MSF output from your PileUp run. Notice the BLOSUM30 matrix specified (others available through the Options menu) and the lowered gap introduction and extension penalties of 5 and 2. Scroll through the alignment then **Close** the window. The next window will be the **Output Manager**. Just as before, click on **Add to Editor** and then specify **Overwrite old with new** in the new **Reloading Same Sequences** window to merge the new alignment with the old alignment and retain all feature annotation. This feature information may help guide your alignment efforts in subsequent steps. **Close** the **Output Manager** window after loading the new alignment.

The alignment should now be a bit better within the specified region. Repeat this process in all areas of low similarity, working from the carboxy termini toward the amino end. Notice that all of the options that were specified are retained by the program. You can also save these run parameters so that they will come up in subsequent sessions by clicking on the **Save Settings** box in any of the program run windows. You may want to periodically go to the **File** menu to save your work using the

Save as . . . function in case of a computer or network problem. It is also a good idea to reperform the PlotSimilarity and color mask procedure after going through the entire alignment to see how things have improved after you have finished the various InSitu PileUps. If you discover an area that you cannot improve through this automated procedure, then it is time to either manually *correct* it or *throw it away.* Note those *problem* areas, then switch back to **Residue Coloring**. This will ease manual alignment by allowing your eyes to work with columns of color.

Other tools that can help manual alignment are **GROUP**ing and **Protections**. The **GROUP** function allows you to manipulate *families* of sequences as a whole. Any change in one will be propagated throughout all. To *GROUP* sequences, select those that you want to behave collectively and then click on the **GROUP** icon right above your alignment. You can have as many groups as required. The space bar will introduce a gap into the sequence and the delete key will take a gap away. However, you can not delete a sequence residue without changing that sequence's or the entire alignment's **Protections**. Click on the padlock icon to produce a **Protections** window. Notice that the default protection allows you to modify **Gap Characters** and **Reversals** only. Check **All other characters** to allow you to *CUT* regions out of your alignment and/or delete individual residues and then click **OK** to close the window. This very powerful manual alignment function can be thought of as the *abacus* function. To take advantage of this function select the region that you want to slide and then press the shift key as you move the region with the right or left arrow key. You can slide residues greater distances by prefacing the command keystrokes with the number of spaces that you want them to slide.

Make subjective decisions regarding your alignment. Is it satisfactory? Do the sequences line up the way that they should? If, after all else, you decide that you just cannot align some region, or even an entire sequence, then perhaps get rid of it with the **CUT** function. Cutting out an entire sequence may leave some columns of gaps in the alignment. If this is the case, then reselect all of your sequences and go to the **Edit** menu and select **Remove Gaps . . . Columns of gaps**. Amino and carboxy termini seldom align properly and are often jagged and uncertain. This is common in multiple sequence alignments and subsequent analyses should probably not include these regions. If loading sequences from a database search, then allowing SeqLab to trim the ends automatically based on beginning and ending constraints considerably improves this situation. Overall, consider strongly conserved residues such as tryptophans, cysteines, and histidines; important structural amino acids such as prolines, tyrosines, and phenylanines, and the conserved isoleucine, leucine, valine triumvirate; make sure they all align. After tweaking, evaluating, and readjusting the alignment, change back to **Feature Coloring Display**. Those features that are annotated should now align perfectly. This is another way to assure that your alignment is as biologically *correct* as possible. Everything you do from this point on, e.g., if you use alignments to ascertain molecular phylogenies, is absolutely dependent on the quality of the alignment! You need a very clean, unambiguous alignment with a very high confidence level in a truly biologically meaningful alignment. Each column of symbols must actually contain homologous characters.

Many other alignment editors are available. However, using a GCG compatible editor like SeqLab, assures that the format will not be corrupted. If you do make any changes to a GCG sequence data file with a non-GCG compatible editor, you must reformat the alignment. Reformatting GCG MSF or RSF files requires a certain level of expertise. If you do need to do this for any reason, you must use the appropriate Reformat option (either -MSF or -RSF, respectively), and you must specify all the sequences within the file using the brace specifier, i.e.,**{*} ,** for example:

```
> reformat -msf your_favorite.msf{*}
```

SeqLab Editor On-Screen Annotation

Adding text annotation to the display and changing the names of the entries for presentation purposes may be helpful. Double-click on an entry's name to reveal the **Sequence Information** window and directly edit the name. Selecting the entry name and then pressing the **INFO** icon can also be used. To put text lines directly into your display go to the SeqLab **File** menu **New sequence . . .** entry and select the **Text** button to the **What type of sequence?** question. This will put a *NewText* line at the bottom of the Editor display allowing the annotation to be inserted. You can also add customized **Graphic Features** and **Features Coloring** annotation with the **Windows Features** window. Select a desired region across an alignment and launch the **Features** window. Press **Add** to get a **Feature Editor** window where you can designate the feature's **Shape: Color:** and **Fill:** as well as giving the region a **Keyword:** and **Comments:**. You can add feature annotation to a region across an entire alignment, but you cannot delete or edit the annotation from the whole region afterwards. You can only edit or delete feature annotation from an RSF file with the SeqLab Editor one sequence feature at a time!

Profile Analysis: Position-Specific, Weighted Score Matrices of Multiple Sequence Alignments

After refining the alignment as much as possible, a powerful approach to suggesting function of distantly related proteins and structural motifs is the Profile suite. This strategy works best when one has prepared and refined a multiple sequence alignment of significantly similar sequences or regions within sequences. A good plan is to identify similar sequences in a newly sequenced section of DNA using traditional database searching techniques and then align all of the significantly similar sequences or domains. Next, run the aligned sequences through the Profile package to generate a profile of the family, a very sensitive probe for further analysis. Searching sequence databases with this probe is tremendously powerful.

Profile methods enable the researcher to recognize features that may otherwise be invisible to individual sequence members. Profile analysis uses the full information content of an alignment. Compared to that of individual sequences, this enhanced information content has the potential to find similar motifs in sequences that are only distantly related. All other methods of describing an alignment such as consensus or pattern description either through away too much information or become too ambiguous. Profiles achieve additional sensitivity with a two-dimensional weight matrix approach, in which conserved areas of the alignment receive the most importance and variable regions the least.

A distinct advantage is that further manipulations and database searches consider evolutionary issues by virtue of their Profile algorithms. The creation of gaps is highly discouraged in conserved areas and occurs easily in variable regions in subsequent profile alignments and searches. This occurs because gaps are penalized more heavily in conserved areas than in variable regions. Furthermore, the more highly conserved a residue, the greater its position-specific matrix score. These two factors give the profiles power. The matrix and its associated consensus sequence are not based merely on the positional frequency of particular residues, but rather utilize the evolutionary conservation of amino acid substitutions within the alignment based on the specified scoring matrix. The BLOSUM62 table is used by default although other substitution matrices can also be specified. Therefore, the resultant consensus residues are the most evolutionarily conserved, rather than just the most statistically frequent.

Traditional Profiles, ala Michael Gribskov

A profile, and its inherent consensus, is created with the GCG program ProfileMake. When a profile is created, all of its members should be appropriately weighted to evenly distribute each contribution. The profile refinement procedure, including repeatedly searching the databases and including or excluding members and adjusting their weights as well as adjusting the profile's length, is known as validating the profile. If you use Profile analysis in your own research, the validation procedures outlined in the GCG Program Manual in the ProfileScan description is prudent. A *motif* style profile library based on the PROSITE Dictionary of Protein Sites and Patterns has been prepared by Gribskov and made available within the GCG system. The program ProfileScan searches the query protein sequence against this library. The present version of GCG has 629 validated profiles in its ProfileScan library.

To run ProfileMake be sure that all of your aligned sequences are selected. Based on your previous observations and your experimental objectives, select the longest, most conserved, sequence length available. Restrict the length of the profile so that the jagged ends in the alignment are excluded. In SeqLab use the **Edit Select Range . . .** menu. **Select** and then **Close** the box. Another effective strategy is to develop multiple shorter profiles centered about the peaks of similarity. These most likely correspond to functional or structural domains. After the range is selected use the **Functions Multiple Comparison ProfileMake** menu and reply **Selected region** in the **Which selection** dialog box. You can also use the **Options. . .** menu from the **ProfileMake** dialog box to specify the **-SeqOut** command option by checking **Write the consensus into a sequence file**, giving it an appropriate name. This will generate a consensus sequence file and a profile file. After running ProfileMake, the top window returned will display the profile consensus sequence. All positions will be filled, because the Profile algorithm selects the most conserved residue for each position. The header contains information relating to the sequence's creation through ProfileMake. **Close** the consensus window. The **Output Manager** will also list a **.prf** file. This is the profile itself which other programs can read and interpret to perform very sensitive database searches and further alignments, using the information within the matrix that penalizes misalignments in conserved areas more than in variable regions. **Save As . . .** the profile giving it an appropriate name that you can recognize and retain the **.prf** extension. **Close** the **Output Manager.**

ProfileSearch is launched from SeqLab with the **Functions** menu; select **Database Sequence Searching ProfileSearch.** Specify the **Query profile. . .** in the **File Chooser** and click **OK.** Search whichever protein database you prefer. ProfileSegments can be run separately after ProfileSearch. The ProfileSearch output file can then be delimited so that ProfileSegments only makes pairwise or multiple alignments of the sequences of interest. Therefore, uncheck **ProfileSegments. . .** to prevent ProfileSearch's output from automatically being passed to ProfileSegments. Under **Options. . . .** use the **-MinList** option by changing **Lowest Z score to report in output list** from 2.5 to 3.5 or higher. -MinList sets a list Z score cut-off value providing a means to limit the output list size. **Close** the **Options** window and be sure that **How: Background Job** is selected and then click **Run.**

As in BLAST and FastA searches, ProfileSearch estimates a significance parameter. In this case it is a Z score based on the distance, that is the number of standard deviations, from the rest of the *insignificant* database matches. ProfileSearch Z scores are normalized and reflect the significance of the results. Rather than randomizing sequences to evaluate a Z score, as in Monte Carlo approaches, it is calculated based on all of the nonsimilar sequences from the database search. As with Monte Carlo approaches, Z scores below 3 are probably not worth considering, from around 4 to 7 may be interesting, and above 7 are most probably significant and should definitely be examined. You can find remote similarities that all other methods will miss using Profile analysis.

Interpreting Profile Analysis

ProfileSearches do require some work to setup and run. They are CPU intensive, and together with HmmerSearch they are some of the most intensive in the GCG package. Be sure to submit them as a batch job as early as possible. If launched from the command line, use the -Batch option. When you examine a ProfileSearch output, take a careful look, there is a good chance that other search algorithms will have missed some of the sequences listed as significant matches. If launched from SeqLab, the output will be located in your working directory and it will have a cryptic name of the form profilesearch_some-number.pfs.

In this example, ProfileSearch finds all of the Elongation Factors in the PIR/NBRF protein database plus other interesting nucleotide binding proteins near the end of the list, all with Z Scores >4. The nucleotide binding motifs in the EF-1α profile are among the most highly conserved portions of the alignment; therefore, greater importance is placed on them by the search. This strategy helps to identify other proteins with similar domains. An abridged screen trace of this ProfileSearch output is shown in Table 5.

The program ProfileSegments constructs BestFit alignments of the results of a ProfileSearch. The MSF option in ProfileSegments allows one to prepare a multiple sequence alignment of the ProfileSearch segments. This can be very helpful for merging ever-increasingly distant sequences into an alignment. The full information content of the profile including the importance of the conserved portions of the alignment is used in this alignment procedure. When running ProfileSegments be sure to set the list size large enough to include all of the relevant sequences in the ProfileSearch output. The -Global option will force full-length alignments, which may be more informative if you are trying to construct a multiple sequence alignment. Figure 11 shows a screen snapshot centered about the t-RNA binding region of a ProfileSegments -MSF -Global alignment, made from the remaining entries in the example in Table 5 aligned against the EF-1α profile.

See companion CD for color Fig. 11

Notice the difference between this alignment and examples seen with other algorithms. Profile alignments often include more gaps than those from other programs. The conserved portions of the profile do not allow the corresponding portion of align-

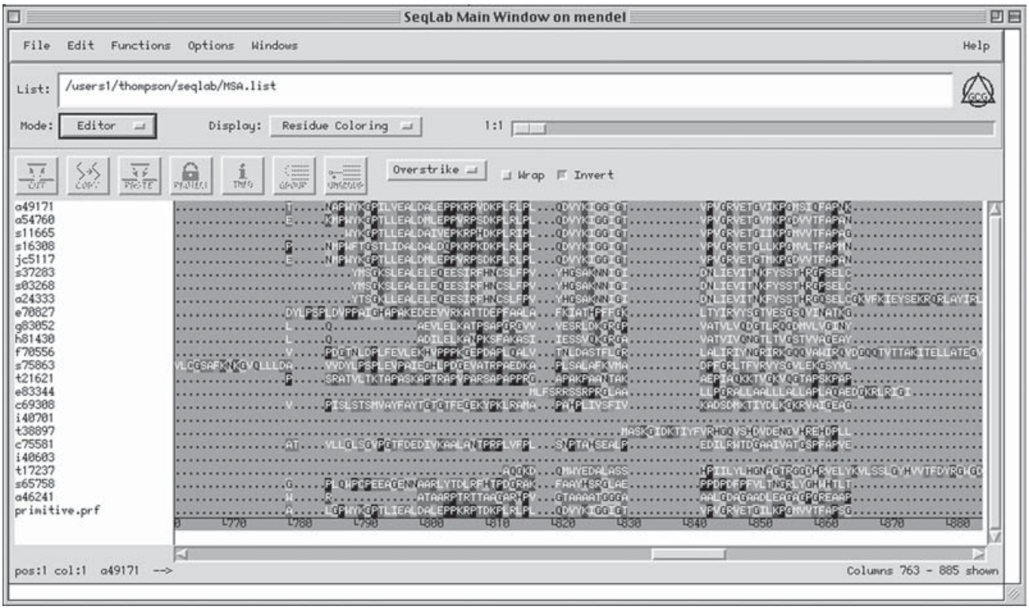
Table 5
ProfileSearch Output, Abridged[a]

```
!!SEQUENCE_LIST 1.0
(Peptide) PROFILESEARCH of: /users1/thompson/seqlab/primitive.prf Length: 428 to: pir:*

         Scores are not corrected for composition effects

                 Gap Weight: 60.00
          Gap Length Weight: 0.67
         Sequences Examined: 188196
         CPU time (seconds): 2713
*    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
Profile information:
(Peptide) PROFILEMAKE v4.50 of:
 @/users1/thompson/.seqlab-mendel/profilemake_63.list Length: 428
 Sequences: 38 MaxScore: 1798.78 July 11, 2001 20:11
                        Gap: 1.00            Len: 1.00
                   GapRatio: 0.33        LenRatio: 0.10
     input_63.rsf{GIARDIA_L} From: 19 To: 446 Weight: 1.00
 input_63.rsf{DIPLOMONAD_SP} From: 19 To: 446 Weight: 1.00 . . .

*    *    *    *    *    *    *    *    *    *    *    *    *    *    *    *
Normalization:                      July 11, 2001 21:21

         Curve fit using 49 length pools
         0 of 49 pools were rejected

         Normalization equation:

                 Calc_Score = 66.96 * ( 1.0 - exp(-0.0023*SeqLen - 0.6191) )

         Correlation for curve fit: 0.973

         Z score calculation:
         Average and standard deviation calculated using 99616 scores
         384 of 100000 scores were rejected

                 Z_Score = ( Score/Calc_Score - 1.010 ) / 0.164

          Sequence Strd ZScore   Orig Length ! Documentation ..
PIR2:A49171           + 158.30 1454.17    435 ! translation elongation factor e
EF-1 alpha chain - Tetrahymena pyriformis
PIR2:A54760           + 157.48 1458.18    449 ! translation elongation factor e
EF-1 alpha chain - Trypanosoma brucei
PIR2:S11665           + 156.90 1458.53    456 ! translation elongation factor e
EF-1 alpha chain - slime mold (Dictyostelium discoid
PIR2:S16308           + 156.81 1449.85    446 ! translation elongation factor e
EF-1 alpha chain - Stylonychia lemnae
PIR2:JC5117           + 155.73 1442.59    449 ! translation elongation factor e
EF-1 alpha - Trypanosoma cruzi
PIR2:T43890           + 154.38 1385.87    395 ! translation elongation factor e
EF-1 alpha [similarity] - Dinenympha exilis (fragmen
PIR2:T43892           + 154.08 1383.28    395 ! translation elongation factor e
EF-1 alpha [similarity] - unidentified Oxymonadida A
PIR2:A60491           + 152.65 1425.02    462 ! translation elongation factor e
EF-1 alpha chain - African clawed frog
PIR2:JU0133           + 152.61 1424.67    462 ! translation elongation factor e
EF-1 alpha chain - Chinese hamster
PIR2:S21055           + 152.61 1424.67    462 ! translation elongation factor e
EF-1 alpha chain - rat
PIR2:I50226           + 152.35 1422.28    462 ! translation elongation factor e
EF-1 alpha - chicken
PIR2:S50143           + 152.24 1421.33    462 ! translation elongation factor e
EF-1 alpha chain - zebra fish
PIR1:EFHU1            + 152.17 1420.67    462 ! translation elongation factor e
EF-1 alpha-1 chain - human
////////////////////////////////////////////////////////////////////////////////
PIR2:S37283           +   9.80 154.03     639 ! tetracycline resistance protein
tetM - Neisseria meningitidis
PIR2:S03268           +   9.80 154.03     639 ! tetracycline resistance protein
tetM - Ureaplasma urealyticum
PIR2:A24333           +   9.69 153.00     639 ! tetracycline resistance protein
tetM - Enterococcus faecalis transposon Tn1545
PIR2:E70827           +   9.66 155.56     701 ! probable fusA protein - Mycobact
erium tuberculosis (strain H37RV)
PIR2:G83052           +   9.66 160.60     840 ! translation initiation factor IF
-2 PA4744 [imported] - Pseudomonas aeruginosa (stra
PIR2:H81430           +   9.44 159.24     871 ! translation initiation factor IF
-2 Cj0136 [imported] - Campylobacter jejuni (strain
PIR2:F70556           +   9.35 149.14     628 ! hypothetical protein Rv1165 - My
cobacterium tuberculosis (strain H37RV)
PIR2:S75863           +   9.28 151.39     691 ! translation elongation factor EF
-G.sll1098 - Synechocystis sp. (strain PCC 6803)
////////////////////////////////////////////////////////////////////////////////
PIR2:S53707           +   4.99 110.33     727 ! translation initiation factor eI
F-2 - bovine
```

Table 5 *(continued)*
ProfileSearch Output, Abridged[a]

```
PIR2:T21621          +   4.98 113.85     878 ! hypothetical protein F32A7.5 - C
aenorhabditis elegans
PIR2:E83344          +   4.98 90.86      317 ! probable adhesion protein PA2407
 [imported] - Pseudomonas aeruginosa (strain PAO1)
PIR2:C69308          +   4.97 93.48      355 ! immunogenic protein (bcsp31-1) h
omolog - Archaeoglobus fulgidus
PIR2:I40701          +   4.97 89.04      294 ! glyceraldehyde-3-phosphate dehyd
rogenase (EC 1.2.1.12) - Citrobacter freundii (frag
PIR2:T38897          +   4.96 82.42      216 ! hypothetical protein SPAC513.02
- fission yeast (Schizosaccharomyces pombe)
PIR2:C75581          +   4.96 105.14     580 ! malate oxidoreductase - Deinococ
cus radiodurans (strain R1)
PIR2:I40603          +   4.96 82.02      212 ! hypothetical protein A - Clostri
dium acetobutylicum
PIR2:T17237          +   4.96 85.14      247 ! hypothetical protein DKFZp434P10
6.1 - human (fragment)
PIR2:S65758          +   4.96 110.29     737 ! nitrate reductase (EC 1.7.99.4)
chain A narB - Oscillatoria chalybea
PIR2:A46241          +   4.95 87.60      277 ! interferon response element-bind
ing factor IREBF-2 - mouse (fragment)
//////////////////////////////////////////////////////////////////////////////
```

[a] A greatly abridged GCG ProfileSearch output list file. Most of the known elongation factors have been edited from the file; several distant homologues are left intact.



List: /users1/thompson/seqlab/MSA.list

Mode: Editor

Display: Residue Coloring

pos:1 col:1 a49171 -->

Columns 763 - 885 shown

Fig. 11. ProfileSegments -MSF -Global output loaded into SeqLab. The t-RNA binding region of a ProfileSegments -MSF -Global alignment of selected near and distant EF-1α homologues aligned against my example EF-1α profile.

ment to gap; yet gaps are easily placed in the nonconserved regions of the alignment. *Clustering* is more Profile analyses than other methods, because of these variable gap penalties. This can be a very useful strategy for pregapping new sequences when introducing them into existing alignments.

HMMER: Hidden Markov Modeling and Profiles

As powerful as the heuristics based Gribskov profiles are, they require a lot of time and skill to prepare and validate. An excess of subjectivity and a lack of formal statistical rigor contribute to their limitations. To address these concerns, GCG incorporated the HMMER (pronounced "hammer") package into GCG version 10.2. HMMER uses Hidden Markov modeling, with a formal probabilistic basis and consistent gap insertion theory, to build and manipulate HMMER profiles and profile databases, to search sequences against HMMER profile databases, and to easily create multiple sequence alignments using HMMER profiles as a *seed.* HMMER profiles are much easier to build than traditional profiles and they do not require as many sequences in their alignments to be effective. They offer a statistical rigor not available in Gribskov profiles and they have all the sensitivity of any profile technique.

Like Gribskov profiles, HMMER profiles are constructed from a set of pre-aligned sequences. However, it is not as important that the alignment be as comprehensive and perfect. To construct a SeqLab HMMER profile alignment, select all of the relevant sequences and perhaps a region within them to exclude, e.g., jagged, unalignable ends. Do not select a Mask sequence, as profiles need to include all of the ambiguity of the alignment within the region being used. Go to the **Functions HMMER** menu and pick **HmmerBuild.** Specify **Selected region** rather than **Selected sequences** if restricting the length of the profile. Accept the default **create a new HMM** and specify an **Internal name for profile HMM.** Specify the **Type of HMM to be Built** as the default is **multiple global**. When the profile is built you need to specify the type of alignment it will be used with, rather than when the alignment is run. The HMMER profile will either be used for global or local alignment and it will occur multiply or singly on a given sequence. Weighting is also considered differently in HMMER than it is with Gribskov profiles. To use a custom weighting scheme, e.g., if you have modified your RSF file weight values for ProfileBuild, select the **-Weighting=N** option. Otherwise HmmerBuild's internal weighting algorithm will calculate the best weights based on a cluster analysis of sequence similarities. It again becomes important to understand the types of biological questions that you are asking to rationally set many of the program parameters.

HmmerCalibrate is checked by default. The completion of HmmerBuild automatically launches a calibration procedure that increases the speed and accuracy of subsequent analyses with the resultant profile. Other HmmerBuild options can be explored, but first read the Program Manual. For now accept the default HmmerBuild parameters and press **Run.** The output is an ASCII text profile representation of a statistical model, a Hidden Markov Model, of the consensus of a sequence family, deduced from a multiple sequence alignment. A utility program, HmmerConvert, can change HMMER style profiles into Gribskov profiles, but information is lost. In most cases the new HMMER profile is used as either a search probe for extremely sensitive database searching or as a template upon which to build ever-larger multiple sequence alignments.

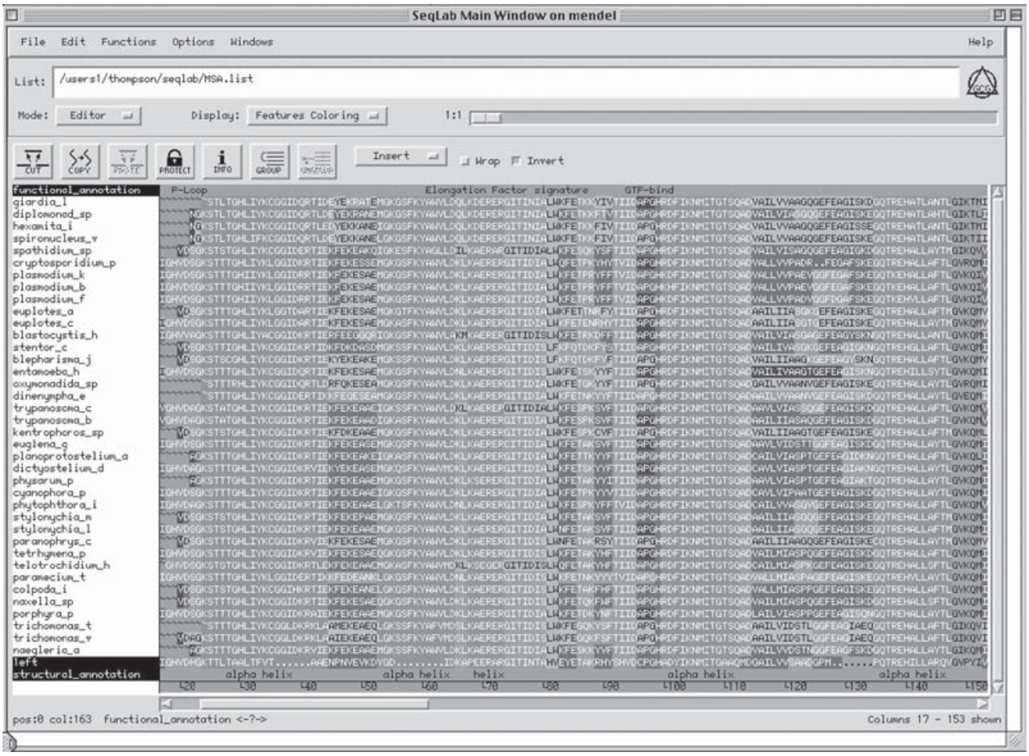
To use the HMMER profile as a search probe go to the **Functions** menu and select **HMMER**, then **HmmerSearch.** Specify the new HMMER profile by clicking **Profile HMM to use as query . . .** and using the **File Chooser** window to select the correct HMMER profile. Either accept the default **Sequence search set . . . PIR:*** specification or choose other sequences to search. HmmerSearch has similar cutoff parameters as other GCG database searches. You can restrict the size of the output based on significance scores to limit the number of pairwise alignments displayed. HmmerSearch is very slow because it is, as a ProfileSearch, a comprehensive, non-heuristic, dynamic programming implementation. So run this program in the background when using SeqLab or, if at a terminal session, use the -Batch command line option. If the server has multiple processors, HmmerSearch supports the multithreading -Processors=x option. **Run** the program when you have the appropriate options selected. The output is extensive and informative, based on significance Expectation-value scores. The top portion is a list of best hits on all domains, the second section is the GCG list file portion of the best domain hits, next pairwise alignments are given, and finally a score distribution is plotted. Since it is a GCG list file, it can be read by other GCG programs, like HmmerAlign.

HmmerAlign can help when working with very large multiple alignments and when adding newly found sequences to an existing alignment regardless of size. Somewhat similar in concept to the -MSF option of ProfileSegments, it takes a specified profile, in this case a HMMER profile, and aligns a specified set of sequences to it, to produce a multiple sequence alignment based on that profile. Unlike ProfileSegments, HmmerAlign takes any GCG sequence specification as input, not just the output from its own database searching program. It is much faster to create very large multiple alignments this way, rather than using PileUp over and over. The rationale being, take the time to make a good small alignment and HMMER profile, then use that to build upon the original. The alignment procedure used by HmmerAlign is a recursive, dynamic programming implementation. It compares the profile's matrix individually against every sequence, until an entire alignment is constructed. HmmerAlign can also use its profile to align one multiple alignment to another and produce a merged result of the two. A heuristic solution is provided in those cases where the original alignment is not one of the two, although optimalization is not guaranteed. To use this option choose **Combine output alignment and ...**, then **another alignment** in the SeqLab **HmmerAlign Options** window. This will launch the command line **-Heuristic=some.msf{*}** option. Aligning the original alignment that was used to construct the profile with another sequence set is very fast and non-heuristic though, using the **-MapAlignment=some.rsf{*}** option. Launch HmmerAlign from the **Functions HMMER** menu by selecting **HmmerAlign.** Specify the correct HMMER profile with the **profile HMM to use...** button and pick the sequences that you want to align to the profile with the **Sequences to align...** button. For example, 1EFT is one of the most similar Elongation Factor 1α homolog to the *lower* eukaryote EF-1α profile example that has a solved structure. Therefore, an alignment of its primary sequence with structural annotation against this sample dataset should allow the inference of secondary structure across the entire alignment. This is the basis of homology modeling. The inferred alpha helices are highlighted in red as shown on the accompanying CD. Figure 12 illustrates the inferred secondary structure based on an alignment of 1EFT, the EF-Tu structure from *Thermus aquaticus*, against the EF-1α HMMER profile.

See companion CD for color Fig. 12

HmmerPfam

As with Motifs and MotifSearch, HmmerPfam can supplement annotations of an RSF file. This program scans sequences against a library of HMMER profiles, by default the Pfam library (A database of protein domain family alignments and HMMs, 1996–2000 The Pfam Consortium). Its output lists Pfam domain matches ranked by



SeqLab Main Window on mendel

List: /users1/thompson/seqlab/MSA.list

Mode: Editor Display: Features Coloring 1:1

pos:0 col:163 functional_annotation <-?->

Columns 17 - 153 shown

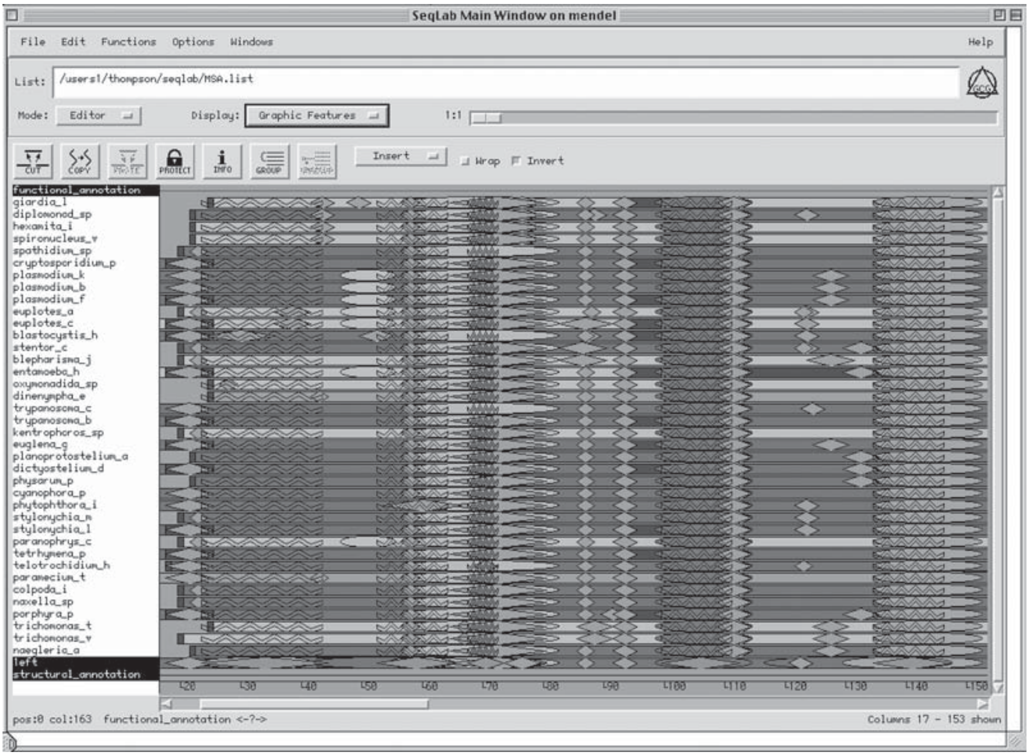
Fig. 12. EF-1α *primitive* dataset aligned to the *Thermus aquaticus* EF-Tu sequence by HmmerAlign. Inferred alpha helices based on the *Thermus* structure are gray in this **Features Coloring** display. Text annotation lines have also been added to the display.

See companion CD for color Fig. 13

Expectation-values and with the -RSF option writes the domain identification and Expectation-value as a feature in an RSF file. The screen snapshot in Fig. 13 shows a sample alignment over the same span as Fig. 12, but includes additional HmmerPfam annotation using **Graphic Features Display:** mode.

Consensus and Masking Issues

Consensus methods are another powerful way to visualize similarity within an alignment. The SeqLab **Edit** menu allows one to easily create several types of consensus representations. To create a standard protein sequence consensus select all sequences and use the **Edit Consensus . . .** menu and specify **Consensus type: Protein Sequence.** When constructing a sequence consensus of a protein alignment you can generate figures with black highly similar residues, gray intermediate similarities, and white non-similar amino acids. The default mode is to create an identity consensus at the two-thirds plurality level (*percent required for majority*) with a threshold of 5 (*minimum score that represents a match*). Different plurality and threshold values as well as different scoring comparison matrices can be used to highlight the difference in the alignment. Be sure that **Shade based on similarity to consensus** is checked to generate a color mask overlay on the display to help in the



SeqLab Main Window on mendel

List: /users1/thompson/seqlab/MSA.list

Mode: Editor

Display: Graphic Features

1:1

□ Wrap ☑ Invert

pos:0 col:163 functional_annotation <-?->

Columns 17 - 153 shown

Fig. 13. Screen snapshot of the sample alignment showing the same region as Figure 12 but now including additional HmmerPfam annotation and displayed with **Graphic Features**. Inferred alpha helices are now seen as transparent medium gray coils.

visualization process. Figure 14 illustrates a region near the carboxy termini using the BLOSUM30 matrix, a *Percent required for majority* (plurality) of 33%, and a *minimum score that represents a match* (threshold) cutoff value of 4.

When you have identified an optimum plurality combination, select the **File Print . . .** command and change the **Output Format:** to **PostScript** to prepare a PostScript file of your SeqLab display. Whatever color scheme is displayed by the Editor will be captured by the PostScript file. As you change the font size the number of pages to be printed varies. In the **Print Alignment** menu specify **Destination . . . File** and give it an appropriate filename and then click **OK.** This command will produce a PostScript language graphics file in the directory from which SeqLab was launched. This PostScript file can be sent to a color PostScript printer, or to a black and white laser printer that will simulate the colors with gray tones, or it can be imported into a PostScript compatable graphics program for further manipulation.

In addition to standard consensus sequences using various similarity schemes, SeqLab also allows one to create consensus *Masks* that screen specified areas of your alignment from further analyses by specifying 0 or 1 weights for each column. A SeqLab Mask allows the user to differentially weight different parts of alignment to reflect confidence. Masks can be modified by hand or they can be created manually

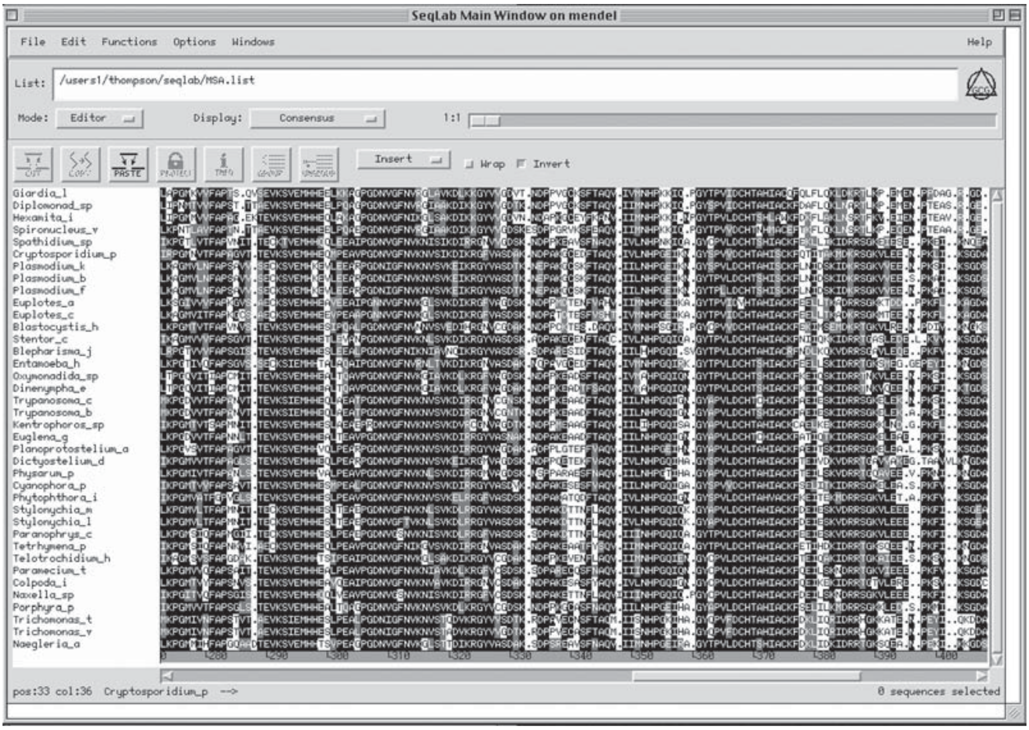


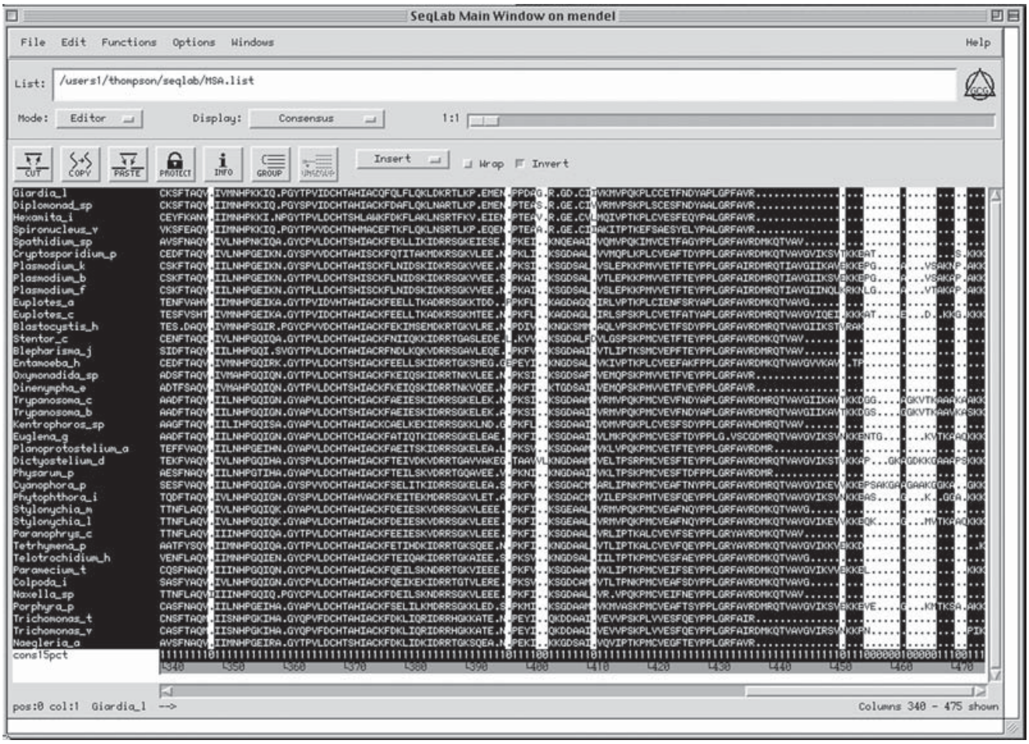
Fig. 14. SeqLab **Consensus** display of a region near the carboxy termini of my EF-1α example using the BLOSUM30 matrix, a 33% **Percent required for majority** (plurality), and a cutoff value of 4 for the **Minimum score that represents a match** (threshold).

using the **New Sequences** menu. They can have position values up to 9. Masking is helpful for phylogenetic analysis as it can exclude less reliable columns in the alignment without actually removing the data. Once a Mask has been created in SeqLab, most of the programs available through the **Functions** menu will use that Mask, if the Mask is selected along with the desired sequences, to weight the columns of the alignment data matrix appropriately. This only occurs through the **Functions** menu.

To create a Mask style sequence consensus select all sequences and then use the **Edit Consensus . . .** menu and specify **Consensus type: Mask Sequence.** As noted earlier, the default mode uses an identity consensus at the two-thirds plurality level with a threshold of 5. These stringent values will likely mask much of the phylogenetically informative data. One can try using lower pluralities, threshold values, and scoring comparison matrices. Figure 15 illustrates the carboxy terminal end using a weight Mask generated from the BLOSUM30 matrix, a plurality of 15%, and a threshold of 4. Few areas are excluded by the Mask in this alignment because of the high similarity of this group of sequences. Excluding more columns would likely leave nearly identical sequences. It would be impossible to ascertain how they are related.

Coding DNA Issues

It is often useful to align DNA sequences along with their corresponding proteins. This is because many investigators prefer to run phylogenetic analyses on DNA rather



List: /users1/thompson/seqlab/MSA.list

Mode: Editor   Display: Consensus   1:1

```
Giardia_l            CKSFTAQV.IVMNHPKKIQ.PGYTPVIDCHTAHIACQFQLFLQKLDKRTLKP.EMEN.PPDAG.R.GD.CIIVKMVPQKPLCCETFNDYAPLGPFAVR......................................
Diplomonad_sp        CKSFTAQV.IIMNHPKKIQ.PGYSPVIDCHTAHIACKFDAFLQKLNARTLKP.EMEN.PTEAS.R.GE.CIVVRMVPSKPLSCESFNDYAALGRFAVR......................................
Hexamita_i           CEYFKANV.IIMNHPKKI.NPGYTPVLDCHTSHLAWKFDKFLAKLNSRTFKV.EIEN.PTEAV.R.GE.CVLMQIVPTKPLCVESFEQYPALGRFAVR......................................
Spironucleus_v       VKSFEAQV.IIMNHPKKIQ.PGYTPVVDCHTNHMACEFTKFLQKLNSRTLKP.EQEN.PTEAA.R.GE.CIIAKITPTKEFSAESYELYPALGRFAVR......................................
Spathidium_sp        AVSFNAQV.IVLNHPNKIQA.GYCPVLDCHTSHIACKFEKLLIKIDRRSGKEIESE..PKEI..KNQEAAI.VQMVPQKIMVCETFAGYPPLGRFAVRDMKQTVAV..............................
Cryptosporidium_p    CEDFTAQV.IVLNHPGEIKN.GYSPVVDCHTAHISCKFQTITAKMDKRSGKVLEE.N.PKLI..KSGDAAL.VVMQPLKPLCVEAFTDYPPLGRFAVRDMKQTVAVGVIKSVTKKEAT.............S.KKK
Plasmodium_k         CSKFTAQV.IILNHPGEIKN.GYSPVLDCHTAHISCKFLNIDSKIDKRSGKVVEE.N.PKSI..KSGDSAL.VSLEPKKPMVVETFTEYPPLGRFAIRDMRQTIAVGIIKAVEKKEPG....A...VSAKNP.AKK
Plasmodium_b         CSKFTAQV.IILNHPGEIKN.GYTPVLDCHTSHISCKFLNIDSKIDKRSGKVVEE.S.PKAI..KSGDSAL.VTLEPKKPMVVETFTEYPPLGRFAIRDMRQTIAVGIIKSVEKKEPG....A...VSAKAP.AKK
Plasmodium_f         CSKFTAQV.IILNHPGEIKN.GYTPLLDCHTSHISCKFLNIDSKIDKRSGKVVEE.N.PKAI..KSGDSAL.VSLEPKKPMVVETFTEYPPLGRFAIRDMRQTIAVGIINQLKRKNLG....A...VTAKAP.AKK
Euplotes_a           TENFVAHV.IIMNHPGEIKA.GYTPVIDVHTAHIACKFEELLTKADRRSGKKTDD..PPKFL..KAGDAGQ.IRLVPTKPLCIENFSRYAPLGRFAVRDMKQTVAVG.............................
Euplotes_c           TESFVSHT.IVMNHPGEIKA.GYTPVIDCHTAHIACKFEELLTKADKRSGKMTEE.N.PKFL..KAGDAGL.IRLSPSKPLCVETFATYAPLGRFAVRDMRQTVAVGVIQEI.KKKAT....E...D..KKG.KKK
Blastocystis_h       TES.DAQV.IVMNHPSGIR.PGYCPVVDCHTAHIACKFEKIMSEMDKRTGKVLRE.N.PDIV..KNGKSMM.AQLYPSKPMCVETFSDYPPLGRFAVRDMRQTVAVGIIKSTVRAK....................
Stentor_c            CENFTAQQ.IVLNHPGQIQA.GYTPVLDCHTAHIACKFNIIQKKIDRRTGASLEDE.L.KVV..KSGDALFDVLGSPSKPMCVETFTEYPPLGRFAVRDMKQTVAV..............................
Blepharisma_j        SIDFTAQV.IILHHPGQI.SVGYTPVLDCHTAHIACRFNDLKQKVDRRSGAVLEQE..PKFV..KSGDAAI.VTLIPTKSMCVEPFTEYPPLGRFAVRDMRQTVAV..............................
Entamoeba_h          CEDFTAQV.IVMNHPGQIRK.GYTPVLDCHTSHIACKFEELLSKIDRRTGKSMEG.GEPEYI..KNGDSAL.VKIVPTKPLCVEEFAKFPPLGRFAVRDMKQTVAVGVVKAV..TP....................
Oxymonadida_sp       ADSFTAQV.IVMAHPGQIQN.GYTPVLDCHTSHIACKFKEIQSKIDRRTNKVLEE.N.PKSI..KSGDSAF.VEMQPSKPMVVETFVEYPPLGRFAVR......................................
Dinenympha_e         ADTFSAQV.IVMAHPGQIQN.GYTPVLDCHTSHIACKFKEIQSKIDRRTNKVQEE.N.PKFI..KTGDSAI.VEMQPSKPMVVETFVEYPPLGRFAVR......................................
Trypanosoma_c        AADFTAQV.IILNHPGQIGN.GYAPVLDCHTCHIACKFAEIESKIDRRSGKELEK.N.PKSI..KSGDAAM.VRMVPQKPMCVEVFNDYAPLGRFAVRDMRQTVAVGIIKAVTKKDGG....AGKVTKAAAKAAKK
Trypanosoma_b        AADFTAQV.IILNHPGQIGN.GYAPVLDCHTSHIACKFAEIESKIDRRSGKELEK.A.PKSI..KSGDAAI.VRMVPQKPMCVEVFNDYAPLGRFAVRDMRQTVAVGIIKAVTKKDGS....GGKVTKAAVKASKK
Kentrophoros_sp      AAGFTAQV.IILIHPGQISA.GYAPVLDCHTAHIACKCAELKEKIDRRSGKKLND.G.PKFL..KSGDAAI.VDMVPGKPLCVESFSDYPPLGRFAVHDMRQTVAV..............................
Euglena_g            AADFTAQV.IILNHPGQIGN.GYAPVLDCHTCHIACKFATIQTKIDRRSGKELEAE..PKFI..KSGDAAI.VLMKPQKPMCVESFTDYPPLG.VSCGDMRQTVAVGVIKSVVKKENTG.......KVTKAAQKKK
Planoprotostelium_a  TEFFVAQV.IILNHPGEIHN.GYAPVLDCHTAHIACKFAEITSKIDRRSGKELEA.L.PKSV..KSGDAAM.VKLVPQKPMCVETFTEYPPLGRFAVRDMR...................................
Dictyostelium_d      TEKFVAQV.IVLNHPGQIHA.GYSPVLDCHTAHIACKFTEIVDKVDRRTGAVVAKEG.TAAVVLKNGDAAM.VELTPSRPMCVESFTEYPPLGRFAVRDMRQTVAVGVIKSTVKKAP...GKAGDKKGAAAPSKKK
Physarum_p           AESFNAQV.IILNHPGTIHA.GYAPVLDCHTAHIACKFTEILSKVDRRTGQAVEE.V.PKNI..KNGDAAI.VKLTPSKPMCVESFTDFPPLGRFAVRDMR...................................
Cyanophora_p         SESFVAQV.IILNHPGQIGA.GYSPVVDCHTAHIACKFSELITKIDRRSGKELEA.S.PKFV..KSGDACM.ARLIPNKPMCVEAFTNYPPLGRFAVRDMRQTVAVGVIKEVVKKEPSAKGAAGAAKGGKA..GKK
Phytophthora_i       TQDFTAQV.IVLNHPGQIGN.GYSPVLDCHTAHVACKFKEITEKMDRRSGKVLET.A.PKFV..KSGDACM.VILEPSKPMTVESFQEYPPLGRFAVRDMRQTVAVGVIKSVVKKEAS....G...K..GGA.KKK
Stylonychia_m        TTNFLAQV.IVLNHPGQIQK.GYAPVLDCHTAHIACKFDEIESKVDRRSGKVLEEE..PKFI..KSGEAAL.VRMVPQKPMCVEAFNQYPPLGRFAVRDMKQTVAVG.............................
Stylonychia_l        TTNFLAQV.IVLNHPGQIQK.GYAPVLDCHTAHIACKFDEIESKVDRRSGKVLEEE..PKFI..KSGEAAL.VRMVPQKPMCVEAFNQYPPLGRFAVRDMKQTVAVGVIKEVVKKEQK....G...MVTKAAQKKK
Paranophrys_c        TTNFLAQV.IIINHPGQIQA.GYTPVLDCHTAHIACKFEEIESKVDRRSGKVLEEE..PKFI..KSGDAAL.VRLIPTKALCVEVFSEYPPLGRYAVRDMKQTVAV..............................
Tetrhymena_p         AATFYSQV.IIMNHPGQIQA.GYTPVLDCHTAHIACKFETIMDKIDRRTGKSQEE.N.PKFI..KNGDAAL.VTLIPTKALCVEVFQEYPPLGRYAVRDMKQTVAVGVIKKVEKKD...................K
Telotrochidium_h     VENFLAQV.IIMNHPGQIEN.GYCPVLDCHTAHIACKFTEIQAKIDRRTGKAIEE.S.PKSV..KNGDSAL.IILTPTKPMCVESFAEYPPLGRFAVRDMKQTVAVG.............................
Paramecium_t         CQSFNAQV.IIINHPGQIQN.GYCPVLDCHTAHIACKFQEILSKNDRRTGKVIEEE..PKFV..KSGDAAM.VKLIPTKPMCVEIFSEYPPLGRFAVRDMKQTVAVGVIKVVEKKEL................KKK
Colpoda_i            SASFYAQV.IVLNHPGQIGN.GYCPVLDCHTAHIACKFQEIKEKIDRRTGTVLERE..PKSV..KSGDCAM.VTLTPNKPMCVEAFSDYPPLGRFAVRDMKQTVAVG.............................
Naxella_sp           TTNFLAQVIIIINHPGQIQ.PGYCPVLDCHTAHIACKFDEILSKNDRRSGKVLEEE..PKFI..KSGDAAL.VR.VPQKPMCVEIFNEYPPLGRFAVRDMKQTVAV..............................
Porphyra_p           CASFNAQV.IILNHPGEIHA.GYAPVLDCHTAHIACKFSELILKMDRRSGKKLED.S.PKMI..KSGDAAM.VKMVASKPMCVEAFTSYPPLGRFAVRDMRQTVAVGVIKSVEKKEVE....G...KMTKSA.AKK
Trichomonas_t        CNSFTAQM.IISNHPGKIHA.GYQPVFDCHTAHIACKFDKLIQRIDRRHGKKATE.N.PEYI..QKDDAAI.VEVVPSKPLVVESFQEYPPLGRFAIR......................................
Trichomonas_v        CASFTAQM.IISNHPGKIHA.GYQPVFDCHTAHIACKFDKLIQRIDRRHGKKATE.N.PEYI..QKDDAAI.VEVVPSKPLVVESFQEYPPLGRFAIRDMKQTVAVGVIRSVVKKPN................PIK
Naegleria_a          AVSFNAQV.IIMNHPGEIRA.GYTPVLDCHTSHIACKFDKLIDKIDRRTGKSQEA.N.PEKI..KKGDSAI.VQVIPTKPMCVEGFTEYPPLGRFAVR......................................
cons15pct            1111111101111111111111111111111111111111111111111111111110111100111111101111111111111111111111111111111111111101110000000010000011100111
                     340       350       360       370       380       390       400       410       420       430       440       450       460       470

pos:0 col:1  Giardia_l  -->                                                                              Columns 340 - 475 shown
```

Fig. 15. SeqLab **Consensus** Mask display of the carboxy terminal region of my EF-1α example using a weight Mask generated from the BLOSUM30 matrix, a plurality of 15%, and a threshold of 4.

than protein, as the multiple substitution models are more robust for DNA. In fact, many phylogenetic inference algorithms do not even take advantage of amino acid similarity when comparing protein sequences; they only consider identities! However, the more diverged a dataset becomes, the more random third and eventually first codon positions become. This introduces noise and error into the analysis. Therefore, often third positions and sometimes first positions are masked. One is always balancing signal against noise. Too much noise or too little signal both degrade the analysis to the point of nonsense.

The logic to this paired protein and DNA alignment approach is as follows:

1. Directly align the DNA. If the DNA sequences are quite similar, then merely create your DNA alignment. Next use the **Edit** menu **Translate** function and the **align translations** option to create the corresponding alignment of the protein sequences. Select the region to translate based on the CDS reference as per each DNA annotation. Be careful of CDS entries that do not begin at position 1; the GenBank CDS feature annotation **/codon_start=** identifies the start position within the first codon listed. You may also have to trim the sequences to contain only the relevant gene, especially if they are genomic. This can be accomplished with appropriate protections (padlock icon). **Group** each protein with its corresponding DNA sequence so that subsequent manipulations will keep them together.

2. Use the protein sequences to create the alignment if the DNA alignment is not obvious. In this case the protein sequences are first used to create an alignment. The corresponding DNA sequences are then loaded. You can find the DNA sequence accession codes in the annotation of the protein sequence entries. Next translate the unaligned DNA sequences into new protein sequences with the Edit-Translate function using the **align translations** option and **Group** these with their corresponding DNA sequences, just as noted earlier. The DNA along with their translated sequences will not be aligned as a set, just the other protein set is aligned. **Group** all of the aligned protein dataset together, but separately from the DNA/aligned translation set. Rearrange your display to place the DNA, its aligned translation and the original aligned protein sequence side-by-side and then manually slide one set to match the other. Use the **CUT** and **PASTE** buttons to move the sequences. When pasting, remember the **Sequence clipboard** contains complete sequence entries, whereas the **Text clipboard** only contains sequence data, amino acid residues or DNA bases. The translated sequence entries can be **CUT** away after they are aligned to the set. Merge the newly aligned sequences into the existing alignment Group. It sounds difficult, but you are matching two identical protein sequences, the DNA translation, and the original aligned protein. The **Group** function keeps everything together so that you do not lose your original alignment as you space residues apart to match their respective codons. Some codons may become spaced apart in this process and will require adjustment. Figure 16 shows a sample dataset in an intermediary stage of alignment. The complete sample data RSF file with all annotation, protein sequences, and DNA sequences is available in the accompanying CD.

See companion CD for color Fig. 16

Multiple Alignment Format and Phylogenetics

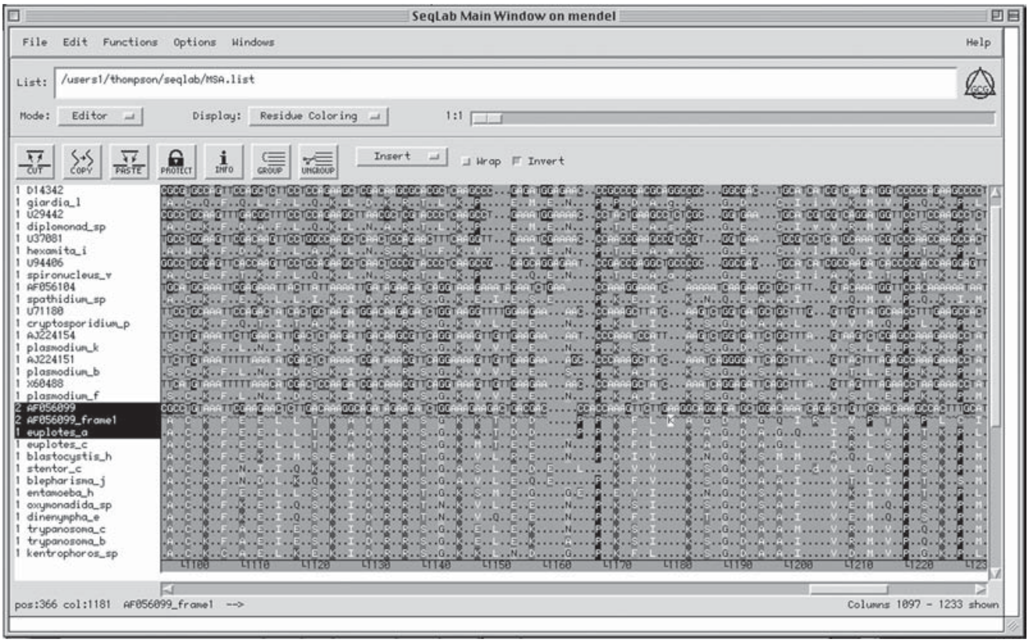
Multiple sequence alignment is a necessary prerequisite for biological sequence based phylogenetic inference, and phylogenetic inference guides our understanding of molecular evolution. The famous Darwinian Theodosius Dobzhansky summed it up succinctly in 1973. It is provided as an inscription on the inner cover of the classic organic evolution text *Evolution*: "Nothing in biology makes sense except in the light of evolution." These words ring true. Evolution provides the single, unifying, cohesive force that can explain all life. It is to the life sciences what the long sought holy grail of the unified field theory is to astrophysics.

*GCG Interface to PAUP**

GCG implements David Swofford's **PAUP*** (usually pronounced "pop star") phylogenetic analysis package with the paired programs **PAUPSearch** and **PAUPDisplay.** These interface programs provide an easy to use access to **PAUP***. However, you may want to consider running PAUP* exterior to GCG by getting the latest version, containing many bug fixes and enhancements, directly from Sinauer Associates (*see* Website: http://www.sinauer.com/Titles/frswofford.htm), and installing it on your own machine. Version 4.0.0d55 PAUP* is included in GCG version 10.3 and runs in native mode or through the PAUPSearch and PAUPDisplay programs. Use the following command in a terminal window to read the license agreement with GCG, if you are curious:

```
> typedata paup-license.txt
```

The PAUP package was originally written to only perform parsimony analysis with either DNA sequences or morphological character data using a Macintosh computer.



List: /users1/thompson/seqlab/MSA.list

Mode: Editor

Display: Residue Coloring

```
1 D14342
1 giardia_l
1 U29442
1 diplomonad_sp
1 U37081
1 hexamita_i
1 U94406
1 spironucleus_v
1 AF056104
1 spathidium_sp
1 U71180
1 cryptosporidium_p
1 AJ224154
1 plasmodium_k
1 AJ224151
1 plasmodium_b
1 X60488
1 plasmodium_f
2 AF056099
2 AF056099_frame1
1 euplotes_a
1 euplotes_c
1 blastocystis_h
1 stentor_c
1 blepharisma_j
1 entamoeba_h
1 oxymonadida_sp
1 dinenympha_e
1 trypanosoma_c
1 trypanosoma_b
1 kentrophoros_sp
```

pos:366 col:1181 AF056099_frame1 -->

Columns 1097 - 1233 shown

Fig. 16. Using SeqLab to align a set of DNA sequences against an already aligned dataset of their translational products.

The package's name changed in version 4.0 by adding the asterisk which means *and other methods,* referring to the incorporation of the minimum evolution distance method and the maximum likelihood method. It was also expanded to a *portable* package capable of being run on many different platforms using a command line interface in addition to its original Macintosh graphical user interface. PAUP* does not include any protein models of evolution other than a crude like/not-like model. Sophisticated protein models can be used by embedding the necessary commands and matrices in the NEXUS file used as input to the package. PAUP*'s DNA models are perhaps the most sophisticated available in any molecular phylogenetic inference software.

NEXUS Format

Within the context of GCG, NEXUS format files are most easily and reliably built from alignments with GCG's own interface to the PAUP* package. PAUPSearch within SeqLab can be used to generate NEXUS format files that can then be directly fed to any version of PAUP*.

Begin NEXUS conversion by selecting all relevant sequences and weight Mask, in the **Main Window** display. Select **PAUPSearch. . .** from the **Functions Evolution** menu to launch the dialogue box. To generate a NEXUS file, run PAUPSearch in its fastest mode without performing a search. Accept the default **Tree Optimality Criterion . . . maximum parsimony** and the **heuristic tree search (fast) . . . Method for Obtaining Best Tree(s).** Be sure that the **perform bootstrap replications. . .** button is not pressed and then launch the **Options** menu by pressing the appropriate button. In the **PAUPSearch Options** menu check the top box to save the PAUPscript file. This is not required for running the programs but is essential to generate a NEXUS file. You can change or leave the file name as you wish. The PAUPscript output file results from the automatic conversion of the alignment to NEXUS format and contains all the PAUP commands as well as their alignment. If needed, the PAUPlog file keeps track of all that happened during the program run and is a good place to look for any error messages. Uncheck the next box, **Perform the analysis.** The program does the conversion and generates the NEXUS script but is prevented from performing the heuristic search for the best tree. This is equivalent to the command line option -NoRun. Scroll through the options menu, leaving the rest of the options at their default settings. **Close** the options menu. Normally PAUP-Search and PAUPDisplay are linked to each other when they are run from the SeqLab interface. Therefore, uncheck the **PAUPDisplay. . .** button in PAUPSearch's main window to turn PAUPDisplay off. Be sure that **How: Background Job** is specified on the main PAUPSearch menu and then press **Run** there. After a moment the output PAUPscript file will be displayed. An abridged protein dataset example is shown in Table 6.

The PAUPscript file contains the NEXUS format file that was generated by GCG to run PAUP*. Notice that the masked columns of the alignment contain zeroes and are excluded from the NEXUS alignment. This file can be used to run the latest version of PAUP*, if it is available. Using a Macintosh may be desirable in order to take advantage of PAUP*'s very friendly Macintosh interface. GCG automatically creates this file, correctly encoding all of the required format data. When using this file as input to native PAUP*, remove any inappropriate commands within the command block near the end of the file with a simple text editor.

PHYLIP Format

Joseph Felsenstein's PHYLogenetic Inference Package (PHYLIP) from the University of Washington (*see* Website: http://evolution.genetics.washington.edu/phylip.html) is a comprehensive freeware suite of thirty different programs for inferring phylogenies that can handle molecular sequence, restriction digest, gene frequency, and morphological character data. Methods available in the package include parsimony, distance matrix, and likelihood, as well as bootstrapping and consensus techniques. A menu controls the programs and asks for options to set and start the computation. Data is automatically read into the program from a text file in a unique PHYLIP format called *infile*. If it is not found, the user types in the proper data file name. Output is written to special files with names like *outfile* and *treefile*. Trees written into *treefile* are in the Newick format, an informal standard agreed to in 1986 by authors of a number of phylogeny packages.

To generate PHYLIP format input from GCG alignments in SeqLab, a combination approach of GCG's ToFastA and Don Gilbert's ReadSeq will be used. First, go to the **SeqLab Main Window File Export** menu; click **Format** and notice that **MSF, GenBank,** and **GDE2.2** are all available for saving a copy of an RSF file in an alternative format. At this point do not export any of these formats. **Cancel** the window. This export routing does not use the Mask data option that would include or exclude columns from your alignment. To take advantage of the Mask data for subsequent phylogenetic analyses, export your alignment using another method. Make sure that all of the relevant sequences, as well as any Mask that you wish to test, are selected. Next, go to the **Functions** menu, where all choices will be affected by the Mask that you have chosen and select **Importing/Exporting ToFastA. . ..** Press **Run** to convert the portion of the alignment that is not masked in the FastA format. FastA is a good inter-

Table 6
An Abridged NEXUS File from PAUPSearch NoRun[a]

```
#NEXUS

[! Aligned sequences from GCG file(s) ‘@/users1/thompson/.seqlab-mendel/paupsear ch_51.list’ ]

[Length: 441  Type: P May 15, 2001 15:07]

[ Name: GIARDIA_L       Len: 441 Check: 2966 Weight: 1.00]
[ Name: DIPLOMONAD_SP   Len: 441 Check: 4608 Weight: 1.00]
[ Name: HEXAMITA_I      Len: 441 Check: 9530 Weight: 1.00]
[ Name: SPIRONUCLEUS_V  Len: 441 Check: 2245 Weight: 1.00]
[ Name: SPATHIDIUM_SP   Len: 441 Check: 2937 Weight: 1.00]
[ Name: CRYPTOSPORIDIUM_P Len: 441 Check: 7665 Weight: 1.00]
[ Name: PLASMODIUM_K    Len: 441 Check: 9956 Weight: 1.00]
[ Name: PLASMODIUM_B    Len: 441 Check: 9937 Weight: 1.00]
[ Name: PLASMODIUM_F    Len: 441 Check: 796 Weight: 1.00]
[ Name: EUPLOTES_A      Len: 441 Check: 8831 Weight: 1.00]
[ Name: EUPLOTES_C      Len: 441 Check: 8653 Weight: 1.00]
[ Name: BLASTOCYSTIS_H  Len: 441 Check: 9014 Weight: 1.00]
[ Name: STENTOR_C       Len: 441 Check: 5386 Weight: 1.00]
[ Name: BLEPHARISMA_J   Len: 441 Check: 7915 Weight: 1.00]
[ Name: ENTAMOEBA_H     Len: 441 Check: 8365 Weight: 1.00]
[ Name: OXYMONADIDA_SP  Len: 441 Check: 8531 Weight: 1.00]
[ Name: DINENYMPHA_E    Len: 441 Check: 5471 Weight: 1.00]
[ Name: TRYPANOSOMA_C   Len: 441 Check: 9945 Weight: 1.00]
[ Name: TRYPANOSOMA_B   Len: 441 Check: 960 Weight: 1.00]
[ Name: KENTROPHOROS_SP Len: 441 Check: 1567 Weight: 1.00]
[ Name: EUGLENA_G       Len: 441 Check: 492 Weight: 1.00]
[ Name: PLANOPROTOSTELIUM_A Len: 441 Check: 8843 Weight: 1.00]
[ Name: DICTYOSTELIUM_D Len: 441 Check: 6233 Weight: 1.00]
[ Name: PHYSARUM_P      Len: 441 Check: 320 Weight: 1.00]
[ Name: CYANOPHORA_P    Len: 441 Check: 4176 Weight: 1.00]
[ Name: PHYTOPHTHORA_I  Len: 441 Check: 804 Weight: 1.00]
[ Name: STYLONYCHIA_M   Len: 441 Check: 2825 Weight: 1.00]
[ Name: STYLONYCHIA_L   Len: 441 Check: 1254 Weight: 1.00]
[ Name: PARANOPHRYS_C   Len: 441 Check: 196 Weight: 1.00]
[ Name: TETRHYMENA_P    Len: 441 Check: 4061 Weight: 1.00]
[ Name: TELOTROCHIDIUM_H Len: 441 Check: 1239 Weight: 1.00]
[ Name: PARAMECIUM_T    Len: 441 Check: 3452 Weight: 1.00]
[ Name: COLPODA_I       Len: 441 Check: 8135 Weight: 1.00]
[ Name: NAXELLA_SP      Len: 441 Check: 6970 Weight: 1.00]
[ Name: PORPHYRA_P      Len: 441 Check: 1559 Weight: 1.00]
[ Name: TRICHOMONAS_T   Len: 441 Check: 6212 Weight: 1.00]
[ Name: TRICHOMONAS_V   Len: 441 Check: 6532 Weight: 1.00]
[ Name: NAEGLERIA_A     Len: 441 Check: 7736 Weight: 1.00]

begin data;
     dimensions ntax=38 nchar=441;
     format datatype=protein interleave gap=.;
     matrix
[                      1                                                    50]
            GIARDIA_L  .......... .......... STLTGHLIYK CGGIDQRTID EYEKRATEMG
        DIPLOMONAD_SP  .......... .......NGK STLTGHLIYK CGGIDQRTLD EYEKRANEMG
           HEXAMITA_I  .......... .......NGK STLTGHLIYK CGGIDQRTLE DYEKKANEIG
       SPIRONUCLEUS_V  .......... .......NGK STLTGHLIFK CGGIDQRTLD EYEKKANELG
        SPATHIDIUM_SP  .......... .....VDSGK STSTGHLIYK CGGIDERTIE KFEKEAKQIG
    CRYPTOSPORIDIUM_P  MGKEKTHINL VVIGHVDSGK STTTGHLIYK LGGIDKRTIE KFEKESSEMG
         PLASMODIUM_K  MGKEKTHINL VVIGHVDSGK STTTGHIIYK LGGIDRRTIE KFEKESAEMG
         PLASMODIUM_B  MGKEKTHINL VVIGHVDSGK STTTGHIIYK LGGIDRRTIE KFEKESAEMG
         PLASMODIUM_F  MGKEKTHINL VVIGHVDSGK STTTGHIIYK LGGIDRRTIE KFEKESAEMG
           EUPLOTES_A  .......... .....VDSGK STTTGHLIYK LGGTDARTIE KFEKESAEMG
           EUPLOTES_C  MGKEKEHLNL VVIGHVDSGK STTTGHLIYK LGGIDARTIE KFEKESAEMG
       BLASTOCYSTIS_H  MGKEKPHINL VVIGHVVAGK STTTGHLIYA CGGIDKRTIE RFEEGGQRIG
            STENTOR_C  .......... .....VDSGK STTIGHLIYK CGGIDKRTID KFDKDASDMG
        BLEPHARISMA_J  .......... .....VDSGK STSCGHLIYK CGGIDKRTIE KYEKEAKEMG
          ENTAMOEBA_H  MPKEKTHINI VVIGHVDSGK STTTGHLIYK CGGIDQRTIE KFEKESAEMG
       OXYMONADIDA_SP  .......... .......... STTTRHLIYK CGGIDQRTLD RFQKESEAMG
         DINENYMPHA_E  .......... .......... STTTGHLIYK CGGIDERTIK KFEQESEAMG
        TRYPANOSOMA_C  MGKEKVHMNL VVVGHVDAGK STATGHLIYK CGGIDKRTIE KFEKEAAEIG
        TRYPANOSOMA_B  MGKEKVHMNL VVVGHVDAGK STATGHLIYK CGGIDKRTIE KFEKEAADIG
      KENTROPHOROS_SP  .......... .....VDSGK STSTGHLIYK CGGIDKRTIE KFDKEAAEMG
            EUGLENA_G  MGKEKVHISL VVIGHVDSGK STTTGHLIYK CGGIDKRTIE KFEKEASEMG
  PLANOPROTOSTELIUM_A  .......... .......AGK STTTGHLIYK CGGIDKRTIE KFEKEAKEIG
      DICTYOSTELIUM_D  MESEKTHINI VVIGHVDAGK STTTGHLIYK CGGIDKRVIE KYEKEASEMG
           PHYSARUM_P  .......... .......AGK STTTGHLIYK CGGIDKRTIE KFEKEAAEMG
         CYANOPHORA_P  MGKQKTHINI VVIGHVDSGK STTTGHLIYK CGGIDKRTIE KFEKEAAEIG
       PHYTOPHTHORA_I  .......... .VIGHVDAGK STTTGHLIYK CGGIDKRTIE KFEKEAAELG
        STYLONYCHIA_M  .......... .....VDSGK STSTGHLIYK CGGIDKRTIE KFEKEPAEMG
        STYLONYCHIA_L  MPKEKNHLNL VVIGHVDSGK STSTGHLIYK CGGIDKRTIE KFEKEAAEMG
        PARANOPHRYS_C  .......... .....VDSGK STTTGHLIYK CGGIDKRVIE KFEKESAEMG
         TETRHYMENA_P  M.GDKVHINL VVIGHVDSGK STTTGHLIYK CGGIDKRVIE KFEKESAEQG
     TELOTROCHIDIUM_H  .......... ...GHVDSGK STSTGHLIYK CGGIDKRTLE KFEKEAAEMG
         PARAMECIUM_T  G.KDKLHVNL VVIGHVDSGK STTTGHLIYK LGGIDERTIK KFEDEANKLG
            COLPODA_I  .......... .....VDSGK STSTGHLIYK CGGIHKRTIE KFEKEANELG
```

(continued)

Table 6 *(continued)*
An Abridged NEXUS File from PAUPSearch NoRun[a]

```
           NAXELLA_SP  .......... .....VDSGK STTTGHLIYK CGGIDKRTIE KFEKESAEQG
           PORPHYRA_P  MGKEKQHVSI VVIGHVDSGK STTTGHLIYK CGGIDKRAIE KFEKEAAEMG
        TRICHOMONAS_T  .......... .......... STTTGHLIYK CGGLDKRKLA AMEKEAEQLG
        TRICHOMONAS_V  .......... .....VDAGK STTTGHLIYK CGGLDKRKLA AIEKEAEQLG
          NAEGLERIA_A  .......... .......AGK STTTGHLIYK CGGIDKRVIE KFEKEAAEMG
//////////////////////////////////////////////////////////////////////////////
[                      401                                        441]
            GIARDIA_L  CCETFNDYAP LGPFAVR... .......... .......... .
        DIPLOMONAD_SP  SCESFNDYAA LGRFAVR... .......... .......... .
           HEXAMITA_I  CVESFEQYPA LGRFAVR... .......... .......... .
       SPIRONUCLEUS_V  SAESYELYPA LGRFAVR... .......... .......... .
        SPATHIDIUM_SP  VCETFAGYPP LGRFAVRDMK QTVAV..... .......... .
    CRYPTOSPORIDIUM_P  CVEAFTDYPP LGRFAVRDMK QTVAVGVIKS VKKE....KK K
         PLASMODIUM_K  VVETFTEYPP LGRFAIRDMR QTIAVGIIKA VKKEAAKNAK K
         PLASMODIUM_B  VVETFTEYPP LGRFAIRDMR QTIAVGIIKS VKKEAAKAAK K
         PLASMODIUM_F  VVETFTEYPP LGRFAIRDMR QTIAVGIINQ LRKNAAKAAK K
           EUPLOTES_A  CIENFSRYAP LGRFAVRDMK QTVAVG.... .......... .
           EUPLOTES_C  CVETFATYAP LGRFAVRDMR QTVAVGVIQE IKKKE.KKKK K
       BLASTOCYSTIS_H  CVETFSDYPP LGRFAVRDMR QTVAVGIIKS TRAK...... .
            STENTOR_C  CVETFTEYPP LGRFAVRDMK QTVAV..... .......... .
        BLEPHARISMA_J  CVEPFTEYPP LGRFAVRDMR QTVAV..... .......... .
          ENTAMOEBA_H  CVEEFAKFPP LGRFAVRDMK QTVAVGVVKA V.TP...... .
       OXYMONADIDA_SP  VVETFVEYPP LGRFAVR... .......... .......... .
         DINENYMPHA_E  VVETFVEYPP LGRFAVR... .......... .......... .
        TRYPANOSOMA_C  CVEVFNDYAP LGRFAVRDMR QTVAVGIIKA VKKDAAAAAK K
        TRYPANOSOMA_B  CVEVFNDYAP LGRFAVRDMR QTVAVGIIKA VKKDGAAVSK K
      KENTROPHOROS_SP  CVESFSDYPP LGRFAVHDMR QTVAV..... .......... .
            EUGLENA_G  CVESFTDYPP LG.VSCGDMR QTVAVGVIKS VKKE.TKAKK K
  PLANOPROTOSTELIUM_A  CVETFTEYPP LGRFAVRDMR .......... .......... .
      DICTYOSTELIUM_D  CVESFTEYPP LGRFAVRDMR QTVAVGVIKS TKKAAAAAKK K
           PHYSARUM_P  CVESFTDFPP LGRFAVRDMR .......... .......... .
         CYANOPHORA_P  CVEAFTNYPP LGRFAVRDMR QTVAVGVIKE VKKEAGKAGK K
       PHYTOPHTHORA_I  TVESFQEYPP LGRFAVRDMR QTVAVGVIKS VKKEG.GGKK K
        STYLONYCHIA_M  CVEAFNQYPP LGRFAVRDMK QTVAVG.... .......... .
        STYLONYCHIA_L  CVEAFNQYPP LGRFAVRDMK QTVAVGVIKE VKKEGTKAKK K
        PARANOPHRYS_C  CVEVFSEYPP LGRYAVRDMK QTVAV..... .......... .
         TETRHYMENA_P  CVEVFQEYPP LGRYAVRDMK QTVAVGVIKK VKKD...... K
     TELOTROCHIDIUM_H  CVESFAEYPP LGRFAVRDMK QTVAVG.... .......... .
         PARAMECIUM_T  CVEIFSEYPP LGRFAVRDMK QTVAVGVIKV VKKE....KK K
            COLPODA_I  CVEAFSDYPP LGRFAVRDMK QTVAVG.... .......... .
           NAXELLA_SP  CVEIFNEYPP LGRFAVRDMK QTVAV..... .......... .
           PORPHYRA_P  CVEAFTSYPP LGRFAVRDMR QTVAVGVIKS VKKEGTKSAK K
        TRICHOMONAS_T  VVESFQEYPP LGRFAIR... .......... .......... .
        TRICHOMONAS_V  VVESFQEYPP LGRFAIRDMK QTVAVGVIRS VKKP....PI K
          NAEGLERIA_A  CVEGFTEYPP LGRFAVR... .......... .......... .

     ;
endblock;

begin paup;
set errorstop;
set criterion=parsimony;
set increase=no;
pset collapse=maxbrlen;
hsearch start=stepwise addseq=simple swap=tbr;
savetrees /brlens file='/users1/thompson/seqlab/paupsearch_51.pauptrees' replace;
quit;
endblock;
```

[a]The PAUPSearch program can reliably and quickly extract NEXUS format from GCG multiple sequence alignments using the -NoRun option. Zero Mask weighted columns are excluded from the file.

mediate format on the way to PHYLIP's required format. The new file will be displayed by SeqLab. The very first part of the protein dataset FastA format output file is shown in Table 7. Notice that it excludes those positions that were masked with zero and that it now follows all FastA format conventions including the automatic conversion of all GCG style gap periods and tildes to the more universal gap hyphen representation. This circumvents the common *dot to dash* problem that is often encountered in sequence format conversion. **Close** the ToFastA output window. You may want to use the **Output Manager** to save the file under a name you recognize using the **Save As . . .** menu.

Table 7
GCG ToFastA Output[a]

```
>GIARDIA_L In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
--------------------STLTGHLIYKCGGIDQRTIDEYEKRATEMGKGSFKYAWVL
DQLKDERERGITINIALWKFETKKYIVTIIDAPGHRDFIKNMITGTSQADVAILVVAAGQ
GEFEAGISKDGQTREHATLANTLGIKTMIICVNKMDDGQVKYSKERYDEIKGEMMKQLKN
IGWK-EEFDYIPTSGWTGDNIMEKSDKMPWYEGPCLIDAIDGLKAPKRPTDKPLRLPIQD
VYKISGVGTVPAGRVETGELAPGMKVVFAPTS-QSEVKSVEMHHEELKKAGPGDNVGFNV
RGLAVKDLKKGYVVGDVTNDPPVGCKSFTAQVIVMNHPKKIQ-PGYTPVIDCHTAHIACQ
FQLFLQKLDKRTLKP-EMENPPDAR-GD-CIVKMVPQKPLCCETFNDYAPLGPFAVR--
-------------------
>DIPLOMONAD_SP In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
-----------------NGKSTLTGHLIYKCGGIDQRTLDEYEKRANEMGKGSFKYAWVL
DQLKDERERGITINIALWKFETKKFTVTIIDAPGHRDFIKNMITGTSQADVAILVIASGQ
GEFEAGISKEGQTREHATLAHTLGIKTLIVCVNKMDDPQVNYSEARYKEIKEEMQKNLKQ
IGYK-DEFDFIPTSGWTGDSIMEKSPNMPWYSGPCLIDAIDGLKAPKRPTDKPLRLPIQD
VYKINGVGTVPAGRVESGLLIPNMTVVFAPST-TAEVKSVEMHHEELPQAGPGDNVGFNV
RGIAAKDIKKGYVVGDTKNDPPVGCKSFTAQVIIMNHPKKIQ-PGYSPVIDCHTAHIACK
FDAFLQKLNARTLKP-EMENPTEAR-GE-CIVRMVPSKPLSCESFNDYAALGRFAVR--
-------------------
>HEXAMITA_I In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
-----------------NGKSTLTGHLIYKCGGIDQRTLEDYEKKANEIGKGSFKYAWVL
DQLKDERERGITINIALWKFETKKFIVTIIDAPGHRDFIKNMITGTSQADVAILVVAAGQ
GEFEAGISSEGQTREHATLANTLGIKTMIVAVNKMDDPQVNYSEARYTEIKTEMQKTFKQ
IGFK-EEFDFVPLSGWTGDNIMEASPKTPWYKGKCLIECIDGLKAPKRPNDKPLRLPIQD
VYKINGVGTVPAGRVESGELIPGMMVVFAPAG-ETEVKSVEMHHEQLAKAGPGDNVGFNI
KGLSAKDIKKGYVVGDVNNDAPKGCEYFKANVIIMNHPKKI-NPGYTPVLDCHTSHLAWK
FDKFLAKLNSRTFKV-EIENPTEAR-GE-CVMQIVPTKPLCVESFEQYPALGRFAVR--
-------------------
>SPIRONUCLEUS_V In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
-----------------NGKSTLTGHLIFKCGGIDQRTLDEYEKKANELGKGSFKYAWVL
DQLKDERERGITINIALWKFETKKFIVTIIDAPGHRDFIKNMITGTSQADVAILVVAAGQ
GEFEAGISKEGQTREHATLANTLGIKTIILCINKMDDPNVNYSKDRYNEIKTEMTKTLVA
IGYK-PEFNYIPTSGWTGLNIMEKTEKTGWYDGPCLIEAIDSLKPPKRPTDKCLRLPIQD
VYKINGVGTVPAGRVESGCLKPNTLAVFAPTN-TAEVKSVEMHHEELPQAEPGDNVGFNV
RGIAAKDIKKGYVVGDSKSDPPGRVKSFEAQVIIMNHPKKIQ-PGYTPVVDCHTNHMACE
FTKFLQKLNSRTLKP-EQENPTEAR-GE-CIAKITPTKEFSAESYELYPALGRFAVR--
-------------------
>SPATHIDIUM_SP In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
----------------VDSGKSTSTGHLIYKCGGIDERTIEKFEKEAKQIGKESFKYAGLL
DILKAERARGITIDIALWKFESQKYSFTIIDAPGHRDFIKNMITGTSQADVAILVISAGQ
GEFEAGIGKDGQTREHALLAYTMGIKQVVVAINKMD-AVQYNEERFTDIKKEVIDYLKK
MGSKKKMLMSLPISGFMGDNLIEKSDKMPWYKGDTILEALDRVERPKRPVAKPLRLPLQD
VYKITGVGTVPVGRVETGVIKPGTLVTFAPVNITTECKTVEMHHQQLEEAIPGDNVGFNV
KNISIKDIRRGNVVGDSKNDPPKEAVSFNAQVIVLNHPNKIQA-GYCPVLDCHTSHIACK
FEKLLIKIDRRSGKEIESE-PKEIKNQEAAIVQMVPQKIMVCETFAGYPPLGRFAVRDMK
```

[a] The GCG ToFastA program reliably converts GCG multiple sequence alignments into Pearson FastA format. This conversion takes advantage of the Mask sequence and changes gap periods to hyphens.

Next, ReadSeq is used to convert this FastA format file to a PHYLIP compatible format. Exit SeqLab with the **File** menu **Exit** choice, or temporarily switch to your background terminal window. If you exit, you will probably be asked if you want to save your RSF file and any changes in your list. Accept the suggested changes with appropriate names and SeqLab will close. This will return you to your terminal window, formerly behind the SeqLab display, where ReadSeq can be run. This program can be used to change your FastA format file into a PHYLIP formatted file. ReadSeq does not allow you to only choose a portion of an alignment, nor does it automatically convert dots and tildes to hyphens. These issues were resolved in SeqLab. Begin ReadSeq by typing **readseq** at your command prompt in the terminal window. ReadSeq first prompts you for an appropriate output file name, not an input file. Do not make a mistake in this step by giving the name of your input file first. If you do, you will overwrite the input file while running the program, and then when it tries to read the input, there will be nothing left to read! Next choose the current PHYLIP format and then designate the input sequence. Do not use the GCG {*} designator; this is not a GCG program. After the program has read all of the input sequences, specify

Table 8
A ReadSeq Screen Trace[a]

```
> readseq
readSeq (1Feb93), multi-format molbio sequence reader.

Name of output file (?=help, defaults to display):
EF1A.phy

         1. IG/Stanford           10. Olsen (in-only)
         2. GenBank/GB            11. Phylip3.2
         3. NBRF                  12. Phylip
         4. EMBL                  13. Plain/Raw
         5. GCG                   14. PIR/CODATA
         6. DNAStrider            15. MSF
         7. Fitch                 16. ASN.1
         8. Pearson/Fasta         17. PAUP/NEXUS
         9. Zuker (in-only)       18. Pretty (out-only)

Choose an output format (name or #):
12

Name an input sequence or -option:
EF1A.tfa
Sequences in EF1A.tfa (format is 8. Pearson/Fasta)
  1) GIARDIA_L In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  2) DIPLOMONAD_SP In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  3) HEXAMITA_I In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  4) SPIRONUCLEUS_V In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  5) SPATHIDIUM_SP In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  6) CRYPTOSPORIDIUM_P In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  7) PLASMODIUM_K In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  8) PLASMODIUM_B In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  9) PLASMODIUM_F In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  10) EUPLOTES_A In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  11) EUPLOTES_C In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  12) BLASTOCYSTIS_H In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  13) STENTOR_C In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  14) BLEPHARISMA_J In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  15) ENTAMOEBA_H In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  16) OXYMONADIDA_SP In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  17) DINENYMPHA_E In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  18) TRYPANOSOMA_C In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  19) TRYPANOSOMA_B In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  20) KENTROPHOROS_SP In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  21) EUGLENA_G In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  22) PLANOPROTOSTELIUM_A In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  23) DICTYOSTELIUM_D In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  24) PHYSARUM_P In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  25) CYANOPHORA_P In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  26) PHYTOPHTHORA_I In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  27) STYLONYCHIA_M In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  28) STYLONYCHIA_L In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  29) PARANOPHRYS_C In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  30) TETRHYMENA_P In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  31) TELOTROCHIDIUM_H In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  32) PARAMECIUM_T In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  33) COLPODA_I In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  34) NAXELLA_SP In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  35) PORPHYRA_P In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  36) TRICHOMONAS_T In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  37) TRICHOMONAS_V In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list
  38) NAEGLERIA_A In situ PileUp of: @/users1/thompson/.seqlab-mendel/pileup_36.list

Choose a sequence (# or All):
all
Name an input sequence or -option:<rtn>
```

[a] A ReadSeq sample screen trace with user responses highlighted in bold.

All the sequences by typing the word **all.** When the program again asks for an input sequence, press return and the file conversion will begin. Table 8 illustrates a sample terminal session screen trace.

The **. . . padded to fit** error message is not cause for concern. However, if a GCG MSF file was used as the input, then an essential change would be required before it would be correct for PHYLIP. The periods and tildes would need to be changed to hyphens (dashes). To make these changes the following UNIX command works well:

```
> tr \~\. \- < infile.phy > outfile.phy
```

Table 9
PHYLIP Format[a]

```
 38 439
GIARDIA_L   ---------- ---------- STLTGHLIYK CGGIDQRTID EYEKRATEMG
DIPLOMONAD  ---------- -------NGK STLTGHLIYK CGGIDQRTLD EYEKRANEMG
HEXAMITA_I  ---------- -------NGK STLTGHLIYK CGGIDQRTLE DYEKKANEIG
SPIRONUCLE  ---------- -------NGK STLTGHLIFK CGGIDQRTLD EYEKKANELG
SPATHIDIUM  ---------- -----VDSGK STSTGHLIYK CGGIDERTIE KFEKEAKQIG
CRYPTOSPOR  MGKEKTHINL VVIGHVDSGK STTTGHLIYK LGGIDKRTIE KFEKESSEMG
PLASMODIUM  MGKEKTHINL VVIGHVDSGK STTTGHIIYK LGGIDRRTIE KFEKESAEMG
PLASMODIUM  MGKEKTHINL VVIGHVDSGK STTTGHIIYK LGGIDRRTIE KFEKESAEMG
PLASMODIUM  MGKEKTHINL VVIGHVDSGK STTTGHIIYK LGGIDRRTIE KFEKESAEMG
EUPLOTES_A  ---------- -----VDSGK STTTGHLIYK LGGTDARTIE KFEKESAEMG
EUPLOTES_C  MGKEKEHLNL VVIGHVDSGK STTTGHLIYK LGGIDARTIE KFEKESAEMG
BLASTOCYST  MGKEKPHINL VVIGHVVAGK STTTGHLIYA CGGIDKRTIE RFEEGGQRIG
STENTOR_C   ---------- -----VDSGK STTIGHLIYK CGGIDKRTID KFDKDASDMG
BLEPHARISM  ---------- -----VDSGK STSCGHLIYK CGGIDKRTIE KYEKEAKEMG
ENTAMOEBA_  MPKEKTHINI VVIGHVDSGK STTTGHLIYK CGGIDQRTIE KFEKESAEMG
OXYMONADID  ---------- ---------- STTTRHLIYK CGGIDQRTLD RFQKESEAMG
DINENYMPHA  ---------- ---------- STTTGHLIYK CGGIDERTIK KFEQESEAMG
TRYPANOSOM  MGKEKVHMNL VVVGHVDAGK STATGHLIYK CGGIDKRTIE KFEKEAAEIG
TRYPANOSOM  MGKEKVHMNL VVVGHVDAGK STATGHLIYK CGGIDKRTIE KFEKEAADIG
KENTROPHOR  ---------- -----VDSGK STSTGHLIYK CGGIDKRTIE KFDKEAAEMG
EUGLENA_G   MGKEKVHISL VVIGHVDSGK STTTGHLIYK CGGIDKRTIE KFEKEASEMG
PLANOPROTO  ---------- -------AGK STTTGHLIYK CGGIDKRTIE KFEKEAKEIG
DICTYOSTEL  MESEKTHINI VVIGHVDAGK STTTGHLIYK CGGIDKRVIE KYEKEASEMG
PHYSARUM_P  ---------- -------AGK STTTGHLIYK CGGIDKRTIE KFEKEAAEMG
CYANOPHORA  MGKQKTHINI VVIGHVDSGK STTTGHLIYK CGGIDKRTIE KFEKEAAEIG
PHYTOPHTHO  ---------- -VIGHVDAGK STTTGHLIYK CGGIDKRTIE KFEKEAAELG
STYLONYCHI  ---------- -----VDSGK STSTGHLIYK CGGIDKRTIE KFEKEPAEMG
STYLONYCHI  MPKEKNHLNL VVIGHVDSGK STSTGHLIYK CGGIDKRTIE KFEKEAAEMG
PARANOPHRY  ---------- -----VDSGK STTTGHLIYK CGGIDKRVIE KFEKESAEMG
TETRHYMENA  M-GDKVHINL VVIGHVDSGK STTTGHLIYK CGGIDKRVIE KFEKESAEQG
TELOTROCHI  ---------- ---GHVDSGK STSTGHLIYK CGGIDKRTLE KFEKEAAEMG
PARAMECIUM  G-KDKLHVNL VVIGHVDSGK STTTGHLIYK LGGIDERTIK KFEDEANKLG
COLPODA_I   ---------- -----VDSGK STSTGHLIYK CGGIHKRTIE KFEKEANELG
NAXELLA_SP  ---------- -----VDSGK STTTGHLIYK CGGIDKRTIE KFEKESAEQG
PORPHYRA_P  MGKEKQHVSI VVIGHVDSGK STTTGHLIYK CGGIDKRAIE KFEKEAAEMG
TRICHOMONA  ---------- ---------- STTTGHLIYK CGGLDKRKLA AMEKEAEQLG
TRICHOMONA  ---------- -----VDAGK STTTGHLIYK CGGLDKRKLA AIEKEAEQLG
NAEGLERIA_  ---------- -------AGK STTTGHLIYK CGGIDKRVIE KFEKEAAEMG

            KGSFKYAWVL DQLKDERERG ITINIALWKF ETKKYIVTII DAPGHRDFIK
            KGSFKYAWVL DQLKDERERG ITINIALWKF ETKKFTVTII DAPGHRDFIK
            KGSFKYAWVL DQLKDERERG ITINIALWKF ETKKFIVTII DAPGHRDFIK
            KGSFKYAWVL DQLKDERERG ITINIALWKF ETKKFIVTII DAPGHRDFIK
            KESFKYAGLL DILKAERARG ITIDIALWKF ESQKYSFTII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWQF ETPKYHYTVI DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ETPRYFFTVI DAPGHKDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ETPRYFFTVI DAPGHKHFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ETPRYFFTVI DAPGHKDFIK
            KGTFKYAWVL DKLKAERERG ITIDIALWKF ETTNRFYTII DAPGHRDFIK
            KASFKYAWVL DKLKAERERG ITIDIALWKF ETENRHYTII DAPGHRDFIK
            KGSFKYAWVL AKMKAERERG ITIDISLWKF ETRKDFFTII DAPGHRDFIK
            KSSFKYAWVL DKLKAERERG ITIDISLFKF QTDKFYSTII DAPGHRDFIK
            KSSFKYAWVL DKLKAERERG ITIDISLFKF QTDKFYFTII DAPGHRDFIK
            KGSFKYAWVL DNLKAERERG ITIDISLWKF ETSKYYFTII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ETGKYYFTII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ETNKYYFTII DAPGHRDFIK
            KSSFKYAWVL DKLKAEREPG ITIDIALWKF ESPKSVFTII DAPGHRDFIK
            KASFKYAWVL DKLKAERERG ITIDIALWKF ESPKSVFTII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ESPKCVFTII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERC ITIDIALWKF ETAKSVFTII DAPGHRDFIK
            KASFKYAWVL DKLKAERERG ITIDIALWKF ETTKYYFTII DAPGHRDFIK
            KQSFKYAWVM DKLKAERERG ITIDIALWKF ETSKYYFTII DAPGHRDFIK
            KGSFKYAWVL DKLKSERERG ITIDIALWKF ETAKYYITII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ETPKYYVTII DAPGHRDFIK
            KTSFKYAWVL DNLKAERERG ITIDIALWKF ESPKYFFTVI DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWKF ETAKSVFTII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDIALWNF ETAKSVFTII DAPGHRDFIK
            KGSFKYAWVL DKLKAERERG ITIDISLWNF ETAKRSYTII DAPGHRDFIK
/////////////////////////////////////////////////////////////////
```

[a]The beginning of the sample dataset in PHYLIP format produced by ReadSeq from a FastA format file. ToFastA stripped zero weight columns and changed gap periods to hyphens reflected in the PHYLIP file.

The first part of the example PHYLIP output file is displayed in Table 9.

Notice that the file begins with two numbers. The first shows the number of sequences in the matrix and the second lists the length of the matrix including any gaps and ambiguities. The next section lists the names of the sequences truncated to ten characters, if necessary, along with all the sequences printed in an *interleaved* fashion. Only the first sequence block lists the names, all others just give the sequence data itself.

Remember to evaluate the terminal ends of the data matrix. If any of the implied *indels* are uncertain (if the sequence lengths were different), then question marks, **?** , are usually more appropriate than hyphens. The hyphens represent gaps caused by an insertion or deletion; this could be misleading. Therefore, edit the output from ReadSeq to replace leading and trailing hyphens in the alignment with question marks or the unknowns characters **n** or **x** depending on which is appropriate. This is also an excellent point to verify that the sequence names are exactly as you wish in the final PHYLIP plots. PHYLIP sequence names can contain very limited punctuation and mixed capitalization and can be up to ten characters in length. Be very careful with these edits so that the alignment does not shift out of phase.

One vital point that can not be repeated often enough is the dramatic importance of your multiple sequence alignments. Subsequent analyses are absolutely dependent upon them. If you are building multiple sequence alignments for phylogenetic inference, do not base an organism's phylogeny on just one gene. Many complicating factors can make interpretation difficult. Unusual phylogenies can result from bad alignments, insufficient data, abjectly incorrect models, saturated positions (homoplasy), compositional biases, and/or horizontal gene transfer. Use several genes, e.g., the Ribosomal Database Project (RDP) (*see* Website: http://rdp.cme. msu.edu/html/) to provide a good, largely accepted, alignment and phylogenetic framework from which other phylogenies can be compared. The complete RDP can be installed on a local GCG server in aligned GCG format. Otherwise desired data subsets can be downloaded from RDP and loaded into SeqLab. Anytime the orthologous phylogenies of organisms based on two different genes do not agree, there is either some type of problem with the analysis, or you have found a case of lateral transfer of genetic material. Paralogous gene phylogenies are altogether another story and if possible should be based on sequences all from the same organism.

There are many situations that will present vexing alignment problems and difficult editing decisions. Most datasets that you will encounter will not have many homologs or related domains, or you will be working on a paralogous system. These are the times that you will really have to think.

Gunnar von Heijne in his quite readable but somewhat dated treatise, *Sequence Analysis in Molecular Biology; Treasure Trove or Trivial Pursuit* (1987), provides an appropriate conclusion:

> "Think about what you're doing; use your knowledge of the molecular system involved to guide both your interpretation of results and your direction of inquiry; use as much information as possible; and do not blindly accept everything the computer offers you."

He continues:

> ". . . if any lesson is to be drawn . . . it surely is that to be able to make a useful contribution one must first and foremost be a biologist, and only second a theoretician . . . . We have to develop better algorithms, we have to find ways to cope with the massive amounts of data, and above all we have to become better biologists. But that's all it takes."

Glossary and Abbreviations

Expectation value (E-value) The likelihood that a particular sequence alignment is due to chance. The value is dependent on sequence and database composition and size and on how often a researcher performs database searches. Most modern sequence database similarity programs such as the BLAST and FastA programs provide this statistic based on the Extreme Value Distribution. The closer the value is to zero, the more significant the match.

Mask In the context of a multiple sequence alignment a mask can be used to create variants of your dataset that ignore certain columns and/or over emphasize other columns of the data, without having to actually duplicate the original dataset and edit it into the desired variants.

Motif A motif is a described and catalogued region of a sequence, usually shorter than a domain, and often, but not always, associated with some biological structural, functional, or regulatory role. Motifs are commonly represented as consensus patterns, but are often described by profiles as well.

Profile A profile is a statistical description of a multiple sequence alignment, commonly of a region or a motif within a multiple sequence alignment. Profiles take many forms associated with the particular programs that create them, e.g. ProfileBuild, HMMer, MEME, but always increase the importance of conserved residues or bases and decrease the importance of variable areas.

Rich Sequence Format (RSF) GCG's proprietary multiple sequence format contains sequence data, sequence names, and sequence annotation, and is read and displayed by their SeqLab Graphical User Interface (GUI). The annotation includes all database reference information, if the sequences come from a database.

Seed In the context of multiple sequence alignments and profiles, a seed refers to a profile that can be used to easily create larger and larger alignments.

Wintel This is a contraction of Microsoft's Windows Operating System (OS) and the Intel brand central processor unit (cpu) that usually exists on computers running that OS.

X-server The X display system is the 'way' that UNIX OS computers pass graphics back and forth. This standard was established back in the early days of computing and seems somewhat backwards in that X windows are 'served' up on your terminal from a 'client' program on the UNIX server computer. Therefore to display X windows on personal, non-UNIX computers, you need to install emulation software for the X-server function.

Z-score and **z-score** The Z-score is based on a normal Gaussian Distribution and describes how many standard deviations a particular score is from the distribution's mean. This is confusingly in contrast to Bill Pearson's z-score in the FastA programs that is a linear regression of the opt score against the natural log of the search set sequence length. The two values, Z and z, have entirely different magnitudes and should not be correlated.

Suggested Reading

SeqLab and Multiple Sequence Reading

Etzold, T. and Argos, P. (1993) SRS—an indexing and retrieval tool for flat file data libraries, Comp. Appl. Biosc. 9, 49–57.

Genetics Computer Group (GCG), a part of Accelrys Inc., a subsidiary of Pharmacopeia Inc. (©1982-2002) Program Manual for the Wisconsin Package, Version 10.2, Madison, WI.

(*see* Website: http://www.accelrys.com/products/gcg-wisconsin-package)

Gilbert, D. G. (1993 [C release] and 1999 [Java release]) ReadSeq, Bioinformatics Group, Biology Department, Indiana University, Bloomington, IN.

(For public domain software, *see* Website: http://iubio.bio.indiana.edu/soft/ molbio/readseq)

National Center for Biotechnology Information (NCBI) Entrez, National Library of Medicine, National Institutes of Health, Bethesda, MD.

(For public domain software, *see* Website: http://www.ncbi.nlm.nih.gov/Entrez)

Smith, S. W., Overbeek, R., Woese, C. R., Gilbert, W., and Gillevet, P. M. (1994) The Genetic Data Environment, an expandable GUI for multiple sequence analysis, Comp. Appl. Biosci. 10, 671–675.

von Heijne, G. (1987) Sequence Analysis in Molecular Biology; Treasure Trove or Trivial Pursuit, Academic Press, San Diego, CA.

Similarity Searching

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990) Basic Local Alignment Tool, J. Mol. Biol. 215, 403–410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25, 3389–3402.

Pearson, W. R. and Lipman, D. J. (1988) Improved tools for biological sequence analysis, Proc. Natl. Acad. Sci. USA 85, 2444–2448.

Pearson, W. B. (1998) Empirical statistical estimates for sequence similarity searches, J. Mol. Biol. 276, 71–84.

Indentifying Motifs

Bailey, T. L. and Elkan, C., (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers, in: Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, CA, pp. 28–36.

Bailey, T. L. and Gribskov, M. (1998) Combining evidence using p-values: application to sequence homology searches, Bioinformatics 14, 48–54.

Bairoch, A. (1992) PROSITE: A dictionary of sites and patterns in proteins, Nucleic Acids Res. 20, 2013–2018.

Gribskov, M., Luethy, R., and Eisenberg, D. (1989) Profile analysis, in: Methods in Enzymology, vol. 183, Academic Press, San Diego, CA, pp. 146–159.

Gribskov, M., McLachlan, M., and Eisenberg, D. (1987) Profile analysis: detection of distantly related proteins, Proc. Natl. Acad. Sciences USA 84, 4355–4358.

The Scoring Matrix

Henikoff, S. and Henikoff, J. G. (1992) Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci. USA 89, 10,915–10,919.

Schwartz, R. M. and Dayhoff, M. O. (1979) Matrices for detecting distant relationships, in: Atlas of Protein Sequences and Structure, vol. 5 (Dayhoff, M. O., ed.), National Biomedical Research Foundation, Washington, DC, pp. 353–358.

Phylogenetics

Dobzhansky, T., Ayala, F. J., Stebbins, G. L., and Valentine, J. W. (1977) Evolution, W. H. Freeman and Co., San Francisco, CA.

The source of the original 1973 quote is obscure though it has been cited as being transcribed from the American Biology Teacher (1973) 35, 125–129.

Felsenstein, J. (1980–2001) Phylogeny Inference Package (PHYLIP) version 3.5+, Department of Genetics, University of Washington, Seattle, WA.

(For public domain software *see* Website: http://evolution.genetics.washington.edu/phylip.html)

Feng, D. F. and Doolittle, R. F. (1987) Progressive sequence alignment as a prerequisite to correct phylogenetic trees, J. Mol. Evol. 25, 351–360

Sogin, M. L., Morrison, H. G., Hinkle, G., and Silberman, J. D. (1996) Ancestral relationships of the major eukaryotic lineages, Microbiolgia Sem. 12, 17–28.

Swofford, D. L. (1989–2001) PAUP* (Phylogenetic Analysis Using Parsimony and other methods), version 4.0+, Illinois Natural History Survey (1994), Smithsonian Institution (1998), Florida State University (2001).

(*see* Website: http://paup. csit.fsu.edu)

Distributed through Sinaeur Associates, Inc., Sunderland, MA.

(*see* Website: http://www.sinauer.com/)

Hidden Markow Models

Eddy, S. R. (1996) Hidden Markov models, Curr. Opin. Struct. Biol. 6, 361–365.

Eddy, S. R. (1998) Profile hidden Markov models, Bioinformatics 14, 755–763

D. Analysis of Gene Expression: *Microarrays and Other Tools*

34 Overview of the Tools for Microarray Analysis

Transcription Profiling, DNA Chips, and Differential Display

Jeffrey A. Kramer

Introduction

In the infancy of the computer revolution of the 1960s, Moore's law, which stated that personal computer speeds would double every 18–24 mo, was coined. A similar law of genomic data could be stated today. The genomic revolution has changed the way that molecular biology is carried out and has necessitated a brand new field of scientific endeavor, bioinformatics. With the increasing number of whole genome sequences being completed, including the recent reports of the completion of the human genome sequence and that of the mouse, it is becoming increasingly possible to study the effects of multiple perturbations on several complex and interconnected gene-signaling pathways. As molecular biology moves to this systems approach to understanding the complex control mechanisms mediating growth, development, and differentiation, the amount of data generated is staggering. Bioinformatics is, among other things, the science of building novel approaches and new tools capable of sorting through vast quantities of genomic data and enabling molecular biologists to draw meaningful conclusions from their data.

The science of genomics can be divided into three main areas, specifically, DNA sequencing, transcription profiling, and proteomics. DNA sequencing is the cornerstone upon which the entire genomic revolution is built. Transcription profiling (TxP) is the study of the response that multiple messenger RNA species present in a given tissue or cell type to specific conditions or treatments. Proteomics is the study of changes in part or all of the protein species present in a given tissue or cell type, including post-translational modifications such as phosphorylation and glycosylation. The three branches of genomics deal with three different macromolecules: DNA, RNA, and proteins. Sequencing aims to unravel the genetic information contained within the genome (DNA). Transcription profiling seeks to understand a tissue or cell's transcriptional response to its environment by identifying all or many of the individual species of messenger RNA and how they change as the cell's environment changes. Proteomics studies the changes in the protein complement present in a specific tissue or cell type under a particular set of conditions. As such, each of the three branches of the science of genomics has specific informatics problems often requiring unique solutions. An understanding of the biological and technical specifics of the questions asked and the answers being sought by scientists in each of the three branches of genomics is necessary to ensure that bioinformatic solutions are of maximum utility. This chapter will attempt to describe the theory and the biology underlying some of the techniques presently being used in transcription profiling.

As detailed earlier, genomic DNA sequencing seeks to identify all of the genes present in a given organism's genome. Bioinformatic approaches have been generated that are capable of predicting genes based solely on genomic sequence data. However, final verification of each predicted gene relies on the identification of a transcription product (a messenger RNA molecule) corresponding to the predicted gene sequence at some point in the lifetime of the organism being studied. Similarly, changes in the occurrence and level of individual mRNA molecules in a particular cell will not always directly correlate with the levels of translation products (proteins) present. Finally, the identity and relative levels of individual proteins present in a cell may not be an adequate predictor of protein activity, as numerous post-translational modifications and protein interactions may effect the activity and localization of proteins within a tissue or cell. For all these reasons, simply identifying all of the mRNA species present, and the levels at which they are present at a particular time, will not yield a complete picture of what is happening inside a particular population of cells. Still, elucidating those mRNA species present and identifying those genes whose expression levels change the most in response to specific conditions or treatments is a useful way to begin to unravel the cellular mechanisms of disease and of drug response. As such, TxP holds the promise of identifying new targets for disease intervention. For this reason the academic community, pharmaceutical and biotechnology industries have embraced transcription profiling as a vital technology.

Transcription profiling, the identification of a profile or fingerprint of some or all of the genes present in a given tissue or cell population at a particular time, need not involve all of the genes in a particular population. Indeed, as the human genome was only recently completed and since the full complement of human genes remains to be determined, true profiling of every human gene in any given experiment is impossible, at least for the time being. However, any attempt to study the transcriptional response of multiple genes can be said to be transcription profiling. As such, there are several experimental approaches to TxP. In general, the experimental methods can be divided into two categories, open and closed. Open methods do not rely on previous knowledge of the sequence or even the identity of the genes that comprise the genome of the organism being studied. Instead, open methods seek to identify mRNA species that demonstrate the most striking or the most interesting response to the experimental conditions utilized and then characterize those species at a sequence level. Closed systems rely on the previous identification and isolation of each RNA species being assayed, and as such, only those genes for which sequence information is available can be studied. Examples of both open and closed methods of transcription profiling and the bioinformatic issues surrounding these activities will be discussed. Differential Display, an example of an open transcription profiling methodology is briefly discussed below. DNA microarrays will be discussed at more length as the primary example of a closed transcription profiling technique.

Differential Display

Open transcription profiling approaches have a clear advantage over closed methodologies since they do not require extensive knowledge of the sequence of the genes that make up the genome. However, these technologies tend to be somewhat labor

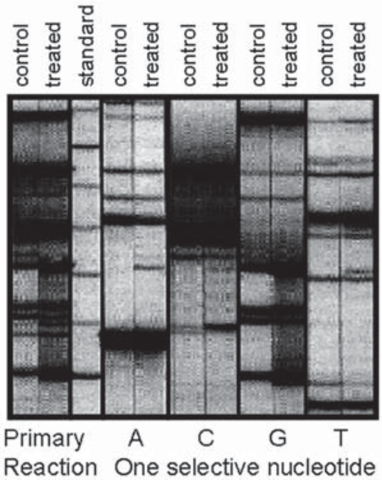


Fig. 1. Polyacrylamide gel electrophoresis is used in differential display to identify fragments of regulated genes. Double-stranded complementary DNA is prepared from populations of RNA, then fragmented with restriction endonucleases. Linkers are attached to the ends of the fragments, then the fragments are amplified using the polymerase chain reaction (PCR). The PCR products are separated by polyacrylamide gel electrophoresis. Differentially expressed genes will yield restriction fragments of different intensity in the treated vs control samples. The primary treated and control samples give a complex mixture of unaffected and differentially expressed fragments. For this reason, fragments are often *parsed* or selectively amplified using one or more degenerate nucleotide at the 3'-position of one or both PCR primers to deconvolute the mixture of amplified species. Addition of a single degenerate nucleotide in one of the PCR primers results in a partial deconvolution of the sample, allowing differentially expressed fragments to be separated and excised more easily.

intensive and are often quite expensive. Perhaps the most widely used open transcription profiling approach is differential display. There are several permutations of differential display, including Celera's Amplified Fragment Length Polymorphism (AFLP)® and Millennium Pharmaceutical's Rapid Analysis of Differential Expression (RADE)™. All differential display methods seek to deconvolute the highly complicated contents of the biological samples being studied in an effort to facilitate the identification of those species most highly affected by the conditions or treatments being studied. Differential display techniques begin with the preparation of RNA from two or more cell populations, such as diseased and normal, or treated and untreated. The first step is often the preparation of double-stranded cDNA. Upon preparation of double-stranded cDNA various methods are employed to identify those species that are differentially expressed. In the differential display technique known as AFLP® the double-stranded cDNA molecule is digested with restriction endonucleases, enzymes that cleave double-stranded DNA at specific sequences. Double-stranded linkers are annealed to the ends of the cDNA molecules and used as priming sites for amplification using polymerase chain reaction (PCR). By using a PCR primer that differs in its final 3'-most nucleotide it is possible to separate the complex mixture of amplified products into four different populations. This tends to reduce the complexity of the amplified products that are subjected to polyacrylamide gel for electrophoresis (*see* Fig. 1). Upon separation by gel electrophoresis, bands that correspond to differentially expressed genes (i.e., those that change in intensity on the gel) are excised, purified, and sequenced. In this way, beginning with no prior knowledge of the sequence of the genes expressed in the biological samples being studied, the researcher generates a sequence of only those genes that are of interest (i.e., those that are differentially expressed). The informatics issues surrounding differential display data analysis are largely the same as the issues surrounding DNA sequencing. That is, the role of the bioinformatician in analyzing differential display data comes close to the end of the technique, and consists largely of database mining and repetitive element masking.

DNA Microarrays

Perhaps the most visible transcription profiling methodology is the DNA microarray, or gene chip. DNA microarrays consist of an array of oligonucleotides or complementary DNA molecules of known composition affixed to a solid support, typically nylon or glass. Gene chips are usually categorized into one of two classes, based on the DNA actually arrayed onto the support. An oligo array is comprised of synthesized oligonucleotides, whereas a cDNA array contains cloned or PCR-amplified complementary DNA molecules. Although the purpose of either type of array is the same, specifically, to profile changes in the expression level of many genes in a single experiment, the two types of arrays present unique bioinformatic challenges. However, most microarray data analysis methods share common themes and goals. As such, the challenges and approaches to data mining are often common between all TxP platforms. The bioinformatics tasks required to complete a thorough analysis of even a single DNA microarray experiment are broken down into four main categories. The four categories are microarray quality control, data visualization and reporting, data analysis, and project organization. Many tools have been developed that simultaneously address several of these categories. For the sake of simplicity, each tool is mentioned only once, but the full utility of each tool may be significantly greater than can be described in a single chapter.

Microarray Quality Control

The two main varieties of microarrays, oligo arrays and cDNA arrays, present different challenges for data QC. These two general classes of array and their unique features will be described in detail in the following. However, regardless of what type of array is used, the data must first pass minimum quality control parameters before any analysis of TxP results can be performed. Microarray experiments involve numerous multi-step procedures at the laboratory bench, both in preparing the actual array and in preparing the labeled probes, as well as hybridizing the probes then washing and reading the microarrays. As such, there are myriad sources of potential error and variability in a microarray experiment and microarrays are often prone to high rates of false-positive and false-negative results. It is therefore vital that all microarray data be passed through several quality control steps. Even after extensive measures are taken to ensure high-quality data, it is imperative that key results be validated by alternative methods. Failure to address these issues will result in significant loss of time and resources spent chasing down numerous false leads. However, as an informatics problem the solution to many of these issues is often trivial. For this reason, the bioinformatician must be adept at applying statistical approaches to ensure that suboptimal quality data is omitted from further consideration. The first-quality control activity after the microarray has been read is to ensure that adequate signal/background and balance was achieved for each array hybridization experiment. This involves an inspection of total average signal and total average background, as well as calculating local signal to background for each element on the array. Elements with signals outside the linear range of the reader, or elements with a very low signal to background should be omitted from further consideration. This is vital to ensure that the resulting data report contains only valid values. Additional quality control metrics are frequently specific for the type of microarray being used. Therefore, a description of the differences between oligonucleotide and cDNA arrays is pertinent.

Oligonucleotide arrays contain short, synthesized oligonucleotides affixed to a solid matrix. Perhaps the best-known commercially available oligonucleotide microarray is the Affymetrix GeneChip®. Individual Affymetrix arrays typically contain 16–20 oligonucleotides for each of up to ~12,000 genes arrayed onto a glass slide approx the size of a microscope slide. Oligos are selected from multiple regions of the gene sequence from which they are derived (*see* Fig. 2). The oligos are typically 25 nucleotides long, and attempts are made in their design to maintain similar GC content, ensuring similar kinetics of binding to the labeled probe. A theoretical difficulty in using short oligonucleotides is the potential for cross-hybridization with other mRNA species. The human genome contains more than 30,000 genes and ~3 billion nucleotides. It is therefore difficult to design short oligonucleotides that are specific to their individual target gene and do not cross hybridize with multiple cDNA probes. To address this, each gene represented on the Affymetrix GeneChip® has several oligos designed to be a perfect match to the target gene to address the potential lack of specificity of short oligonucleotides. Additionally, each perfect match oligo has a paired oligo designed to be identical to the perfect match oligo except for a single nucleotide mismatch (*see* Fig. 2). Many of the mismatched probes and indeed some of the perfect-match probes may cross-react with closely related cDNA species. If a mismatched probe gives a significant signal, both it and its perfect match partner may be excluded from consideration. With the potential for cross reactivity in mind, oligonucleotides that are arrayed are designed to regions of low sequence conservation with genes that are closely related to the target gene. Additionally, Affymetrix array analysis considers signal intensity and signal to background levels for every perfect match/mismatch pair spotted onto the array. Pairs of perfect match/mismatch elements that give significantly more signal than the mismatch element are considered to have a *high average difference* (*see* Fig. 3). Pairs with approximately the same degree of signal are referred to as *low average difference* and a *negative average difference* refers to pairs in which the mismatch oligonucleotides signal is greater than that of the perfect match oligonucleotides. Only those perfect match elements with relatively high signal intensities, and a high average difference are used to make a *present* or *absent* call. As well as investigating each element, data concerning all of the perfect match/mismatch pairs that correspond to a gene are considered. Minimum requirements with respect to the positive/negative ratio (pairs where the perfect match is greater than the mismatch vs those where the mismatch is greater), positive fraction, and log average signal ratio are considered. Those elements that do not meet minimum signal intensity and cutoffs are referred to as *absent*. The signal from these elements is typically not considered in the subsequent analyses. These elements that meet the requirements can be evaluated against other arrays, to assess *increase*, *decrease* or *no change* in signal.

See
companion CD
for color Fig. 3

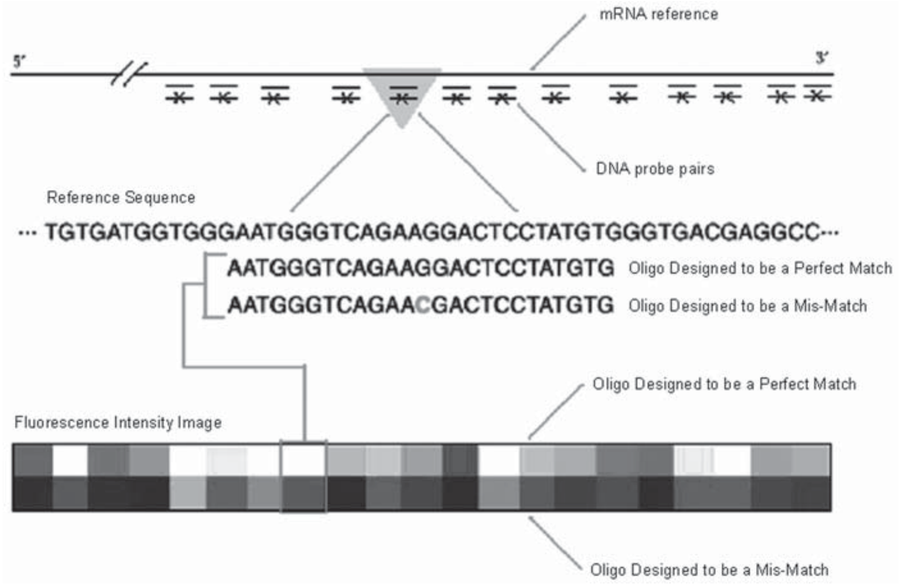

mRNA reference
5'
3'
DNA probe pairs
Reference Sequence
··· TGTGATGGTGGGAATGGGTCAGAAGGACTCCTATGTGGGTGACGAGGCC···
AATGGGTCAGAAGGACTCCTATGTG Oligo Designed to be a Perfect Match
AATGGGTCAGAACGACTCCTATGTG Oligo Designed to be a Mis-Match
Oligo Designed to be a Perfect Match
Fluorescence Intensity Image
Oligo Designed to be a Mis-Match

Fig. 2. Affymetrix uses several oligonucleotides to get broad coverage of individual genes. Each gene is represented by oligos designed to be a perfect match to the target sequence, as well as oligos designed to contain a single mismatch. Pairs of perfect match and single nucleotide mismatch oligonucleotides are designed to different regions of each gene. Typically 16 or 20 oligo pairs are arrayed for each gene represented on the GeneChip®. The oligonucleotides are typically 25 bases in length.

Because the oligonucleotides are designed to be similar with respect to length, GC content and melting temperature (T_M), the kinetics of binding of the labeled probe are highly reproducible from one element to the next. For this reason, oligonucleotide arrays are often referred to as *single-channel arrays*. That is, a labeled cDNA population from a single biological source can be hybridized onto each microarray. Because the kinetics of hybridization are effectively identical for all of the elements on an oligonucleotide array, samples hybridized onto separate arrays can be compared with only minimal mathematical signal balancing. In this way, each individual control sample, and each treated or affected sample is used to prepare a labeled probe, then hybridized onto an individual microarray. This means that within each individual array, it is theoretically possible to compare one element (for example an element corresponding to acyl CoA oxidase, or ACOX) with each other element (such as one corresponding to fatty acid synthase, or FAS). The signal at one element relative to any other element may contain information about the environment and response of the biological sample from which the probe was generated. For example, a liver from a rat treated with a novel chemical entity might have an increased level of ACOX and decreased level of FAS signal relative to an untreated rat liver. The increase in ACOX and the concomitant decrease in FAS, might suggest something about the new compound to an investigator. Additionally, the values at each element, after some minimal balancing based upon signal intensity of a number of control elements, can be compared to each matched element on a second microarray. The values that result from these comparisons can be expressed a number of ways.

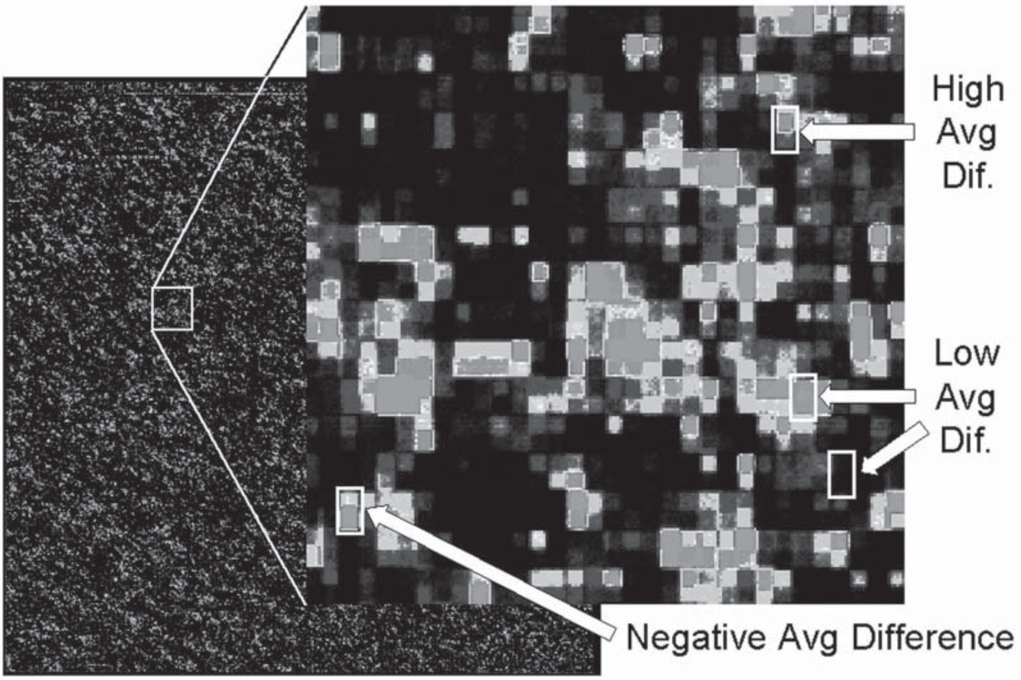


Fig. 3. False color images demonstrate signal intensity on an Affymetrix GeneChip®. Signal intensity at each element is indicated by color, with lighter, *hotter* colors representing greater signal intensities. Many single-channel, oligonucleotide arrays, including Affymetrix arrays, include several oligonucleotides for each gene. For this reason, there tend to be clusters of elements with similar signal intensity, representing all the elements that comprise a single gene. In the expanded region of the array, individual Perfect match/mismatch paired elements can be visualized. Perfect match/mismatch pairs for which the signal intensity of the perfect match oligo is greater than for the mismatch oligo (high average difference pairs), are used when comparing arrays to make calls regarding differential expression.

Unlike oligonucleotide arrays, cDNA array elements contain longer cloned or PCR amplified cDNAs. Often the length of the cDNA species arrayed onto a cDNA array varies considerably. Additionally, the GC content varies, and the T_M can vary significantly from element to element. For this reason, the time required for the hybridization of a labeled probe onto the microarray to come to a steady state is often greater than the stability the labeled probes would allow. For this reason cDNA arrays tend to be hybridized for preset durations according to standard operating procedures. As a result of the short hybridization times relative to steady state, most species of labeled cDNA in the probe do not adhere to steady state binding kinetics, and the binding at each element varies according to the T_M of the probe specific for each element. Because of this, the signal at any element on an individual cDNA microarray cannot be reliably compared with an other element on that array in a meaningful way. To continue the previous example, if the signal at the ACOX element is greater than that for the FAS element, it may be a function of the composition and T_M of the two arrayed elements and not due the amount of each message present in the labeled probe.

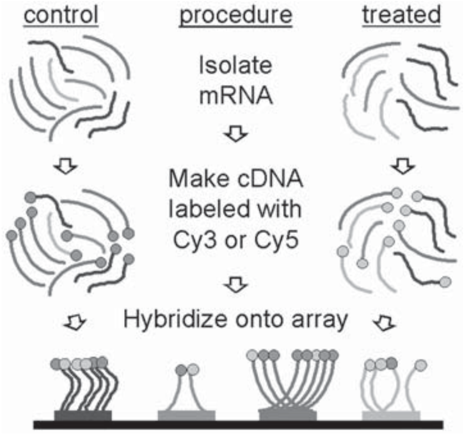


Fig. 4. Experiments using cDNA microarrays require two-channel hybridizations. Two channel or cDNA arrays contain cloned or PCR amplified fragments of genes. Two separate populations of RNA are labeled with different fluorescent dyes. The dye-labeled samples are applied to the array, and differential hybridization is measured by recording fluorescence in both channels at each element. Single-channel signal intensity at any element on a cDNA array may not accurately reflect the amount of the message present in the original biological sample relative to any other message. Instead, the ratio of Cy5 vs Cy3 labeled cDNA probe at each element contains information regarding differential expression.

See companion CD for color Fig. 4

The way to address this issue is through the use of differential or 2-channel hybridization. Two populations of mRNA (i.e., treated and untreated, diseased vs normal, and so on) are labeled and hybridized onto an individual cDNA microarray. A control sample is used to prepare cDNA labeled with one fluorescent label (such as Cy3), and a treated or affected sample is labeled with an alternate fluor (such as Cy5). Both labeled cDNA populations are then mixed and applied to the microarray for hybridization (*see* Fig. 4). If there was twice as much of a particular RNA species present in the Cy5 labeled probe as in the Cy3 labeled probe, twice as much of the Cy5 probe will hybridize to the corresponding element on the array, regardless of the T_M of the arrayed element or the probes. Thus, although the single-channel data in either the Cy5 or the Cy3 channel on a cDNA array may not be informative, the ratio of signal intensities in both channels is. For this reason, cDNA array experiments are often referred to as *2-channel hybridizations*.

See companion CD for color Fig. 5

The main deliverable for the biologist using microarrays is information regarding the differential expression of genes. This differential expression is determined by comparing two oligo arrays, or the two channels of a cDNA array. In order to compare the two channels in a 2-channel array, or to compare two single-channel arrays, the signal intensities must first be balanced. The simplest approach to balancing uses the total average signal in one channel divided by the total average signal in the other channel to generate a balance coefficient. This can be demonstrated graphically using an expression histogram (*see* Fig. 5), where the two lines on the histogram represent either the two channels in a cDNA array experiment, or two single-channel oligo arrays. Slight differences in the amount of probe, or in the labeling reaction efficiency can result in higher total signal intensities on one channel than on another.

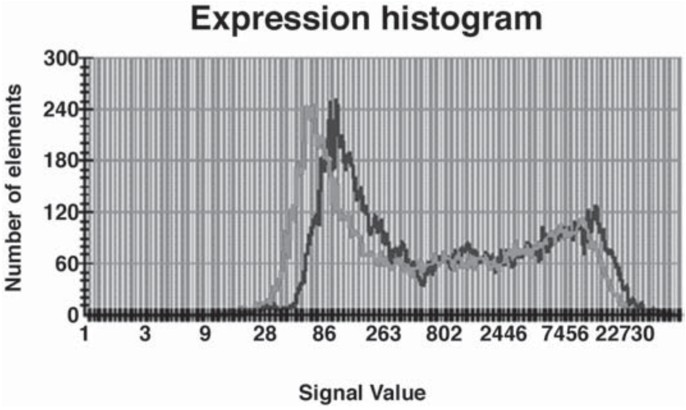


Fig. 5. An expression histogram from a two-channel hybridization reveals how well the channels are balanced. An expression histogram is a convenient way to visualize differences in input material and/or labeling efficiency for a two-channel hybridization. After balancing the array data using a balance coefficient, the two lines should largely overlap. Balance coefficients can be determined in a number of ways. The simplest way to calculate a balance coefficient relies on a ratio of total average signal in both channels.

A balancing coefficient will even out such experimentally derived sources of error. The balance coefficient is used to correct the signal in one channel (or on one array) at all of the elements on the array. If the total average signal were 2500 in the Cy3 channel, and 2000 in the Cy5 channel, the balance coefficient would be 1.25 (2500/2000). The Cy5 signal for each element would be multiplied by 1.25 to generate a balanced Cy5 signal. Alternatively, the Cy3 signal at each element could be corrected by a factor of 0.8 (2000/2500) to generate a balanced Cy3 signal.

Other more complex methods of balancing 2-channel arrays have been used. One difficulty with simply using total average signal is that it ignores the biology of the samples being assayed. For example, if a scientist is comparing an untreated cell population with a population of cells treated with a potent transcriptional activator, the treated cell population may have significant increases in the levels of many genes. The increased signal in the treated sample may be due to real and significant increases in transcription. Applying a balance coefficient based solely upon signal intensity may over *correct* many of the biologically relevant changes in expression. Additionally, balancing simply by total average signal fails to account for changes in signal and signal to background across a microarray. If the signal and the background in both channels are similar and evenly distributed across the entire array (*see* Fig. 6), a simple balancing approach is adequate. However, if the background intensity in the two channels is very different, or not evenly distributed, other approaches may be needed. For example, it is not unusual to see differences in signal and background due to different labeling efficiencies and gradient effects. Figure 7 shows an example of a 2-channel array with both a gradient effect and a large difference in signal and background between the two channels. Although this array experiment has significant issues, there may still be some useful data present. However, care must be taken before trusting any expression data resulting from this array.

See companion CD for color Figs. 6 and 7

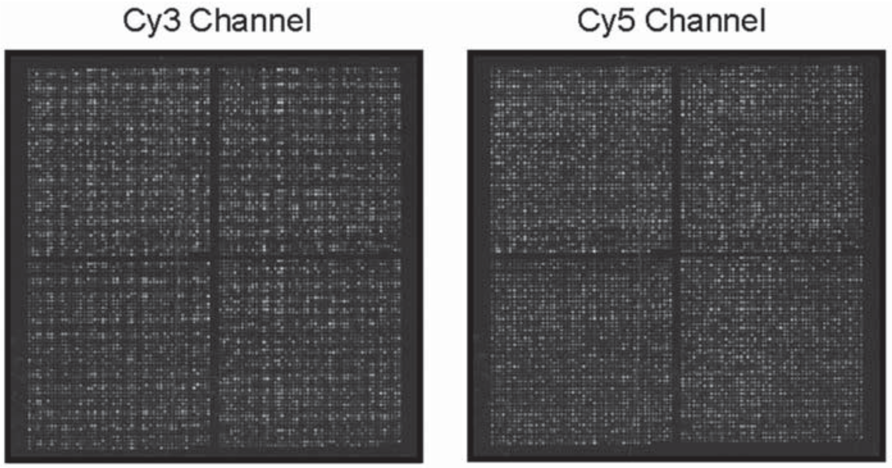


Fig. 6. False color images can be used to demonstrate signal intensity in both channels of a two-channel array hybridization. Signal intensity at each element in either channel is indicated by color, with lighter, *hotter* colors representing greater signal intensities. After balancing, elements that give greater intensity in one channel than in the other are considered to represent differentially expressed genes.

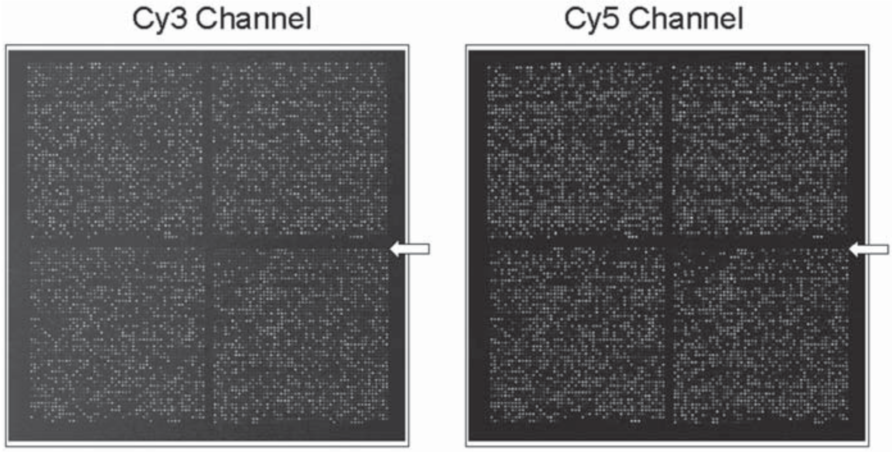


Fig. 7. Unacceptable hybridization on a two-channel array is determined by careful examination of several parameters. In some cases, total average signal intensity, total average background, or both are very different between the two channels of a two-channel hybridization. If the background is uniform in each channel or if a gradient is present in both channels, it can often be corrected. If, as in the example shown, there is a gradient effect in one channel and not the other, it may be very difficult to achieve reliable expression data from the microarray.

First, note that there appears to be a gradient in the background of the Cy3 channel that is not present in the Cy5 channel. That is, the background is greater on the left side of the array. Simply balancing based on the total average signal ignores the fact that the higher background on the left part of the array may cause the signal intensities of the elements on that portion of the array to be affected differently than elements on the right side of the array. Balancing elements individually by signal to

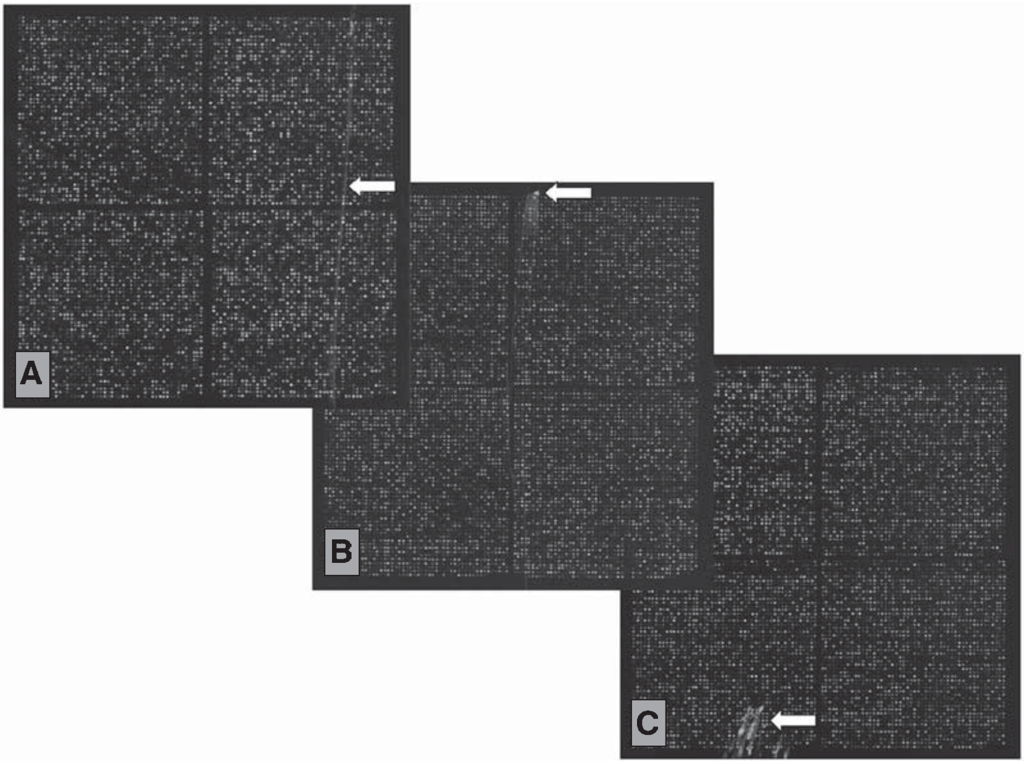


Fig. 8. Imperfections and impurities that affect hybridization and therefore expression data may occur on microarrays. **(A)** Some microarrays may contain imperfections that occur during the fabrication process. **(B)** Elements can also be affected by impurities introduced at the time of the hybridization. The *comet-like* imperfection seen here is most likely due to a dust particle. **(C)** It is uncertain what caused the impurity on the lower portion of this array. It may be an artifact from the scanning process. Regardless of the cause, the affected elements may need to be discarded from further consideration.

local background may address this problem. An additional problem with this array is that the balance coefficient will be unusually large. A very large or a very small balance coefficient may bring seemingly meaningful signals from noise, or mute signals beyond the linear range of the reader used to read the hybridized array. However, re-running this experiment may not be an option owing to the prohibitive costs often associated with running many microarrays. Additionally, there may be some useful and reliable information in this experiment. For example, the arrows point to an element that may be differentially expressed in the Cy5-labeled population when compared to the Cy3-labeled population. Therefore, new approaches to glean useful data from such an experiment, without providing false or misleading data is a key challenge.

See companion CD for color Fig. 8

Another type of problem that often occurs on both single and 2-channel array experiments is caused by imperfections or impurities on the array. Figure 8 shows several examples. In Fig. 8A, the line running across the array will change the signal to background of elements along its course. Additionally, elements along the course of the line may appear to have a signal that is significantly above the global average background. The array in Fig. 8B appears to contain a bit of dust on the top of the array. The elements in the *tail* of the comet-like feature will be affected similarly to those near the linear imperfection in Fig. 8A. The array in Fig. 8C has an unusual pattern of unknown cause. The affected elements should be omitted from further consideration. Such impurities may occur on either single-channel or 2-channel arrays. If they are faint and if they occur similarly in both channels of a 2-channel array, they might be corrected for. But in general, the elements affected by these features should usually be discarded from the final data report.

Software Tools for Microarray Analysis

Several proprietary software tools exist that allow scientists to select individual elements or groups of elements for masking from further analysis. However many of these tools tend to be designed to work with only one type of array. A general tool capable of working with all types of arrays that a laboratory might use has yet to be made available. The Institute for Genome Research (TIGR) (*see* Website: http://www.tigr.org/), has a software tool called the *TIGR Spotfinder* available to academic users at no charge. TIGR Spotfinder is a PC/Windows compatible software tool written in C/C++. It was designed for Microarray image processing using 16-bit gray-scale TIFF image files generated by most microarray scanners. Spotfinder allows for manual and automatic grid drawing, and spot detection can be adjusted for use with arrays with multiple size spot and spacing. One particularly useful feature of the TIGR Spotfinder software tool is that it allows the user to adjust the grid to be sure that an entire spot is included in the reference field (*see* Fig. 9). It is not uncommon for spots on a microarray to be off center, and spot finding software using a static grid may often cut off portions of spots. Spotfinder provides a local background correction for each spot, and calculates ratios using one of several user-preferred criteria. One other interesting feature of the TIGR Spotfinder software is the ability to remove saturated pixels from intensity calculations. This feature helps calculate accurate differential expression values for extremely abundant messages, the fluorescent signal for which may be beyond the linear range of the detector. Finally, Spotfinder allows the data to be exported as an excel file and into a format compatible with other TIGR software tools.

The National Human Genome Research Institute (NHGRI) microarray project, (*see* Website: http://www.nhgri.nih.gov/) is also developing a microarray analysis tool, called *ArraySuite*. This suite is a collection of extensions written for IPLab, an image-processing package by Scanalytics (*see* Website: http://www.scanalytics.com/). The NHGRI approach to microarray image analysis is divided into several discreet tasks, including array segmentation and spot identification, background intensity determination, target detection and signal-intensity determination, and ratio calculation. The various capabilities being designed into the ArraySuite tool set will also permit ratio analysis, a comparison of the intensities of a number of *housekeeping* genes, as well as multiple image analysis. The NHGRI is also working on tissue arrays, a method of arraying up to 1000 fixed tissue samples onto a glass slide for high throughput *in situ* analysis of DNA, RNA or protein. This novel approach requires additional new bioinformatic approaches to data acquisition and analysis.

Data Visualization and Reporting

After the quality of the microarray data has been confirmed, the data must be reported in a way that facilitates the rapid determination of significant results. This is a significant challenge, as the amount of data provided by a single microarray

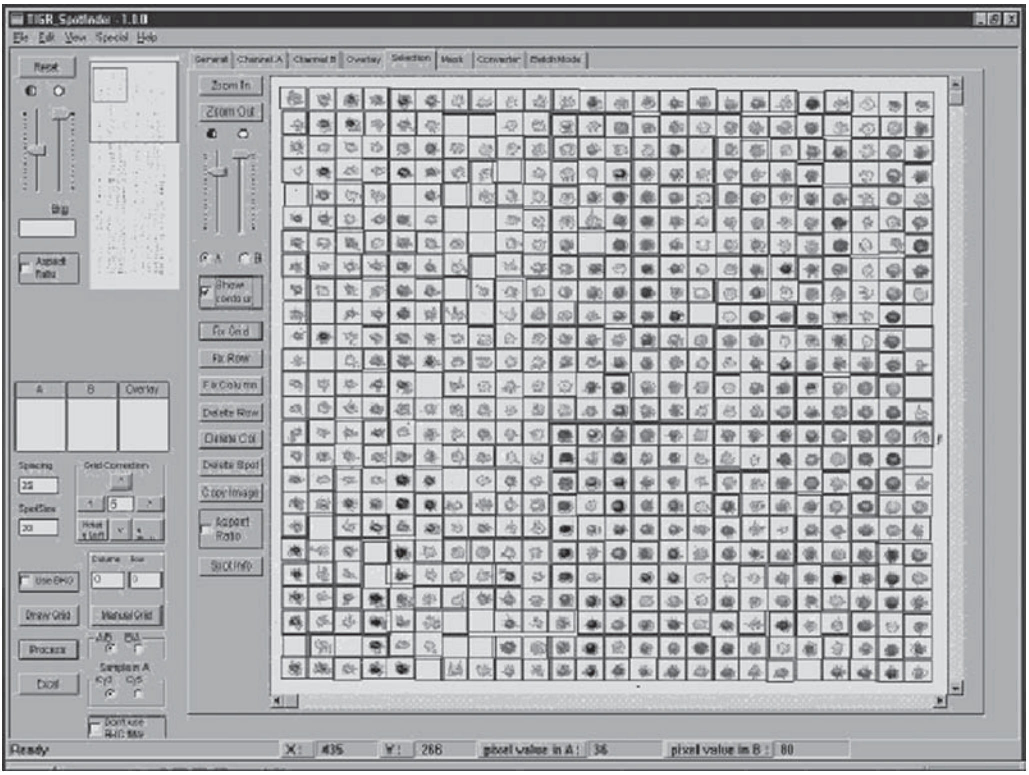


Fig. 9. The TIGR Spotfinder tool allows the user to adjust the grid to be sure that an entire spot is included in the reference field. It is not uncommon for spots on a microarray to be off center. Spot finding software that uses a static grid may often cut off portions of spots, resulting in a loss of useful information. Additionally, badly off-center spots may result in a portion of one spot appearing in the grid of an adjacent spot, resulting in *false-positive* gene expression values.

experiment is very large. Simply reporting the data as a list of signal intensities and element IDs is inadequate. As microarrays are prone to error, it is often desirable to verify microarray findings using other molecular biological methodologies. Data must be expressed in a way that can be compared to results from other molecular techniques. Typically, the purpose of running microarray experiments is to identify new targets for intervention into disease; new molecular markers and predictors of disease, toxicity, or clinical outcome; or a better understanding of molecular mechanisms. In all of these applications, the purpose for the experiment cannot be met by simply providing a list of regulated genes. However, as a first step, it is necessary to express the results in a format that is both intuitive and amenable to further analysis. Most array analysis software packages calculate differential expression values from the elements that pass quality-control parameters. Several methods of relating differential expressions exist. The simplest is a ratio of treated sample intensity divided by control intensity for each element. If the Cy3 channel of a 2-channel array contained the control sample probe and the Cy5 channel contained the treated sample probe, the Cy5 signal intensity is divided by the Cy3 signal. One difficulty with the

Table 1
Differential Expression

Gene ID	*Control (Cy3)*	*Treated (Cy5)*	*bCy5*	*Ratio*	*Log ratio*	*BDE*	*fBDE*
a	500	800	1000	2.00	0.30	2.00	1.00
b	500	200	250	0.50	–0.30	–2.00	–1.00
c	500	100	125	0.25	–0.60	–4.00	–3.00
d	500	80	100	0.20	–0.70	–5.00	–4.00
e	500	2000	2500	5.00	0.70	5.00	4.00
f	500	404	505	1.01	0.00	1.01	0.01
g	505	400	500	0.99	0.00	–1.01	–0.01
h	250	600	750	3.00	0.48	3.00	2.00
i	2500	3800	4750	1.90	0.28	1.90	0.90

expression ratio is obvious when the data is graphed. Down regulated genes tend to be collapsed between 0 and 1 on the graph, making it difficult to distinguish among down regulated genes. This is easily alleviated by using a log scale when graphing, however the degree of down regulation is not as intuitive as using simple ratios. Some researchers use an arithmetic log of the expression ratio instead of a simple ratio. A difficulty with this approach is that significantly regulated genes yield deceptively small log ratio values (*see* Table 1). Another way to report differential expressions is with the use of the balanced differential expression value (BDE). The BDE is calculated using Equation 1, assuming that Cy5 is the treated and Cy3 is the control sample.

If Cy5   Cy3, then BDE = Cy5/Cy3, else BDE = –1 × (Cy3/Cy5)   [Eq. 1]

A difficulty with the BDE value is that it leaves a gap between 1 and –1. That is, it is impossible to calculate a BDE of 0. This causes a problem when trying to compare expression values with phenotypes, or when trying to calculate a meaningful average BDE for a group of similarly affected or similarly treated biological samples. Additionally, two elements that are effectively identical may appear to be significantly different from one another (*see* example genes *f* and *g* in Table 1). For this reason, some bioinformatic scientists use a fractional Balanced Differential Expression (fBDE) value, essentially is a BDE collapsed by 1. It is calculated using Equation 2 where:

fBDE = (Cy5 – Cy3)/MIN (Cy5, Cy3)   [Eq. 2]

the Cy3 (control sample signal intensity) is subtracted from the Cy5 treated sample intensity), and this value is divided by the smaller of the two values. Table 1 shows several theoretical control and treated signal intensities, and the values calculated using these four methods of showing differential expression. The hypothetical data in Table 1 is first corrected using a balance coefficient of 1.25 to generate a balanced Cy5 (bCy5) signal intensity.

Another issue that must be considered when reporting microarray data is the significance of fold induction values. Simply stating a differential expression value does not reveal anything about the reliability of that value. For example, genes *d* and *e* are both regulated fivefold. However, the signal intensity for the two channels (or on the

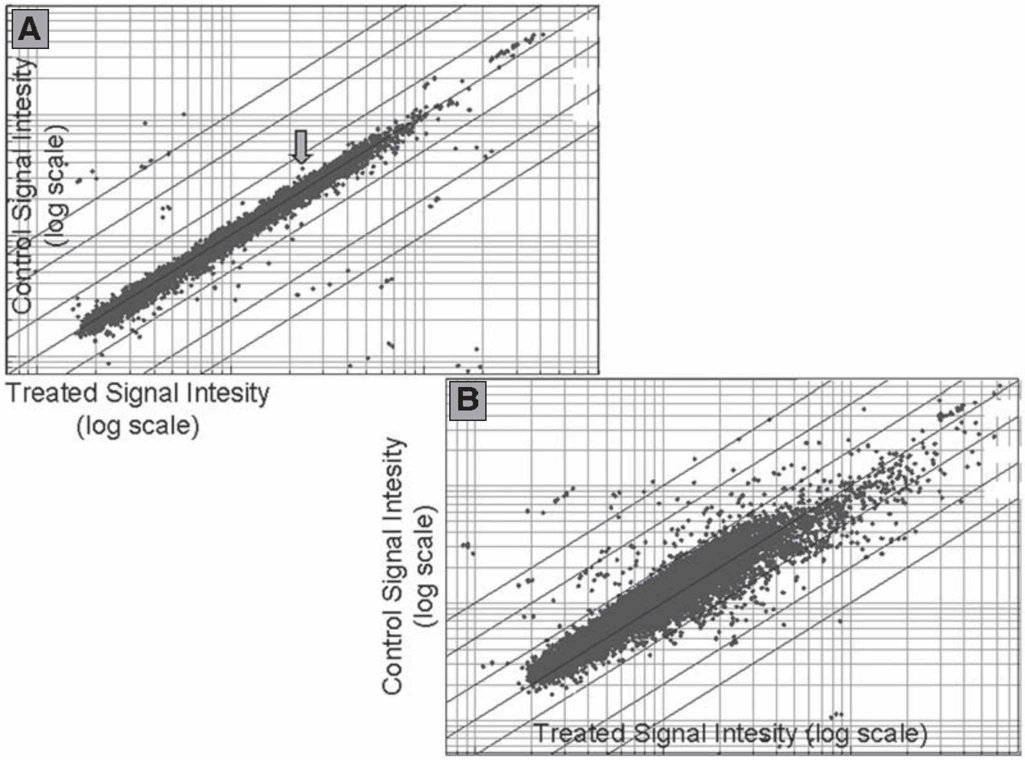


Fig. 10. Scatter plots can be generated from a two-channel hybridization or from any two single-channel hybridizations, after data is balanced with respect to signal intensity. Signal intensity for each element from 2 oligo arrays, or from both channels of a cDNA array is graphed on a logarithmic scale. Genes lying on the slanted line with a slope of one are not regulated. Those elements that demonstrate differences in signal intensity in the two channels (or on two different oligo arrays) after balancing may be differentially expressed. The differential expression value for the element marked in **(A)** may be reliable, even though it represents only a 1.6-fold difference in the signal intensities of the two samples. A similar differential expression value in **(B)** would almost certainly be meaningless.

two arrays if this were an oligonucleotide array experiment) for gene *e* is significantly higher than for gene *d*. Similarly, although gene *i* has a smaller differential expression value than gene *h*, the signal intensity at element *i* is strong, whereas that for element *h* is more questionable. This issue can be envisioned by examining a scatter plot from two microarray experiments (*see* Fig. 10). In the examples shown, the first scatter plot shows very tight clustering of elements around the 1 X line. The second scatter plot shows significant deviation from the 1 X line. This does not mean that useful data cannot be attained from the second array. However, the cut-off of reliable differential expression values will be different for the two experiments. The arrow in Fig. 10A points to a gene with a balanced differential expression value of approx 1.6. This gene is clearly separated from the background *noise* of unregulated genes, and the value of 1.6 can be trusted with a good degree of comfort. Statistically, the confidence that this value represents an actual gene regulation event is strong. In the experiment shown in Fig. 10B, a BDE of 1.6 is clearly not significant. If a researcher were to give attention

See companion CD for color Fig. 10

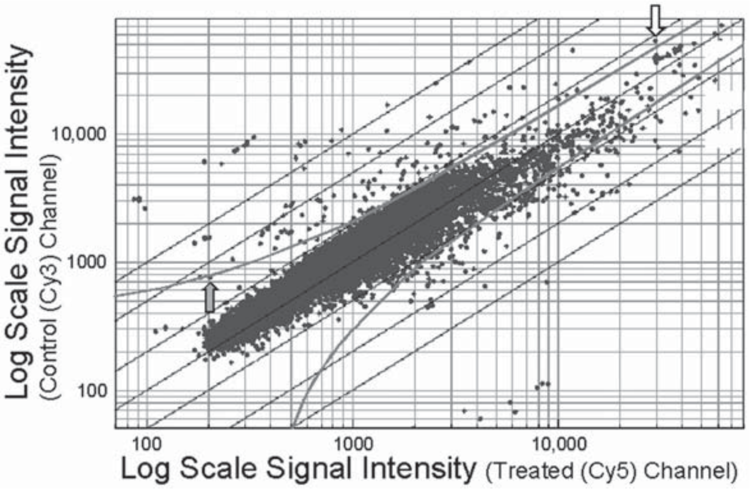


Fig. 11. Confidence limits may be placed on microarray data. The lines parallel to the diagonal represents a reasonable cut-off based upon signal strength and differential expression. Elements with lower signal intensity must be highly differentially expressed to be considered reliable, owing to the increased error at the lower edge of the signal range. RNA quality, time since fabrication of the array, hybridization conditions and numerous other factors can all affect the final outcome of a microarray experiment. Reproducibility and reliability of differential expression values are affected by these variables (*see* Fig. 9). Therefore, confidence limits should be adjusted to reflect better or worse hybridizations.

to every gene regulated at 1.6-fold or more, the list of regulated genes would be difficult to manage. Statistical methods to calculate meaningful expression values for each individual microarray experiment would provide valuable confidence limits for the expression values. Among the criteria that must be addressed in such a statistical confidence limit are the signal intensity and the ratio of signal to background. Simply reporting a BDE does not reveal the strength of the data upon which the expression value is based. Figure 11 shows a graphical demonstration of one approach to this issue. Any points within the two lines closest to the diagonal are not considered to be significantly different. For example, the element marked with a gray arrow is not considered reliable, due to low signal intensity, even though the BDE is about four-fold. However, the element marked with a white arrow may reflect a legitimate change in expression of that gene. Although the BDE is only about 1.8, the high signal intensity suggests a more reliable result. There are several statistical methods to generate confidence limits. As described earlier and as demonstrated in Fig. 10, it is important that the methods be flexible, to reflect scatter plots with more or less noise.

See
companion CD
for color
Fig. 11

Upon calculating meaningful expression values, there is still the problem of visualizing all of the data in an intuitive way. Several approaches to this problem exist. One simple way is to port the data into a spreadsheet and color cells in the spreadsheet based upon the expression values. A brief inspection of the spreadsheet informs the viewer of up and down regulated genes (*see* Fig. 12). The scientist can then zoom in to see the identity of the gene whose expression pattern stands out. Of course, it is very difficult to view such a spreadsheet generated on a single page from even a modestly

See
companion CD
for color
Fig. 12

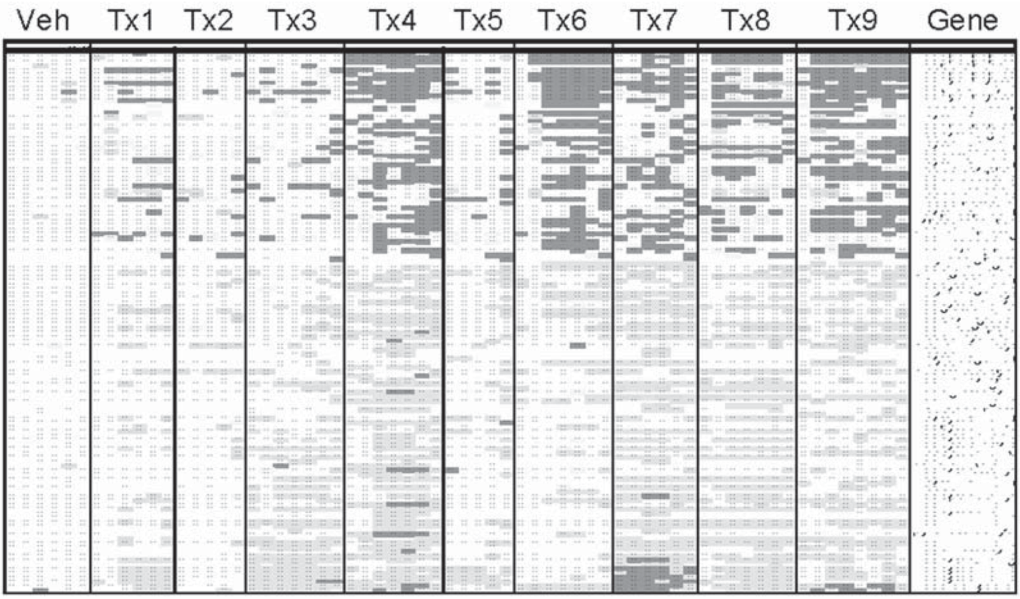


Fig. 12. The application of color gradients to differential expression values aids in rapid visualization of differential expression. The use of color gradients also permits the identification of genes that are differentially regulated by multiple treatments, and may also help identify experimental outliers. In the example shown, data from several microarrays hybridized with RNA from cells treated with vehicle (*veh*) and one of nine compounds (*Tx1–9*) is used. This approach can be used with a simple spreadsheet. However, many software tools exist that facilitate this type of data visualization.

See companion CD for color Fig. 13

sized TxP experiment. For this reason, other commercially available tools such as Spotfire™ have been created specifically for this purpose (*see* Fig. 13). Many of these tools also have data analysis features. However assigning a color gradient, with intensity or brightness corresponding to signal intensity or BDE is essentially the basis of how these data viewers work.

The TIGR Array Viewer is a software tool designed to accommodate both presentation and analysis of microarray data. Array Viewer has the ability to use data from flat files or from an array database and provides links between the elements to the underlying gene information. The software allows for the selection of a subset of elements, for example differentially expressed genes, or genes from a specific region of the array. The subset of genes can be visualized independently, or with the rest of the data but in different colors. This feature is useful if a researcher suspects regional variations on an array. For example, the elements on the left side of the array shown in Fig. 6 can be selected and colored differently from the elements on the right side of the array on a scatter plot. If all or most of the differentially regulated genes are from one or the other region, it is immediately obvious that there are subarray variations and that the BDE values may not be reliable. Although simple inspection of the pseudo color image in Fig. 7 would have revealed this, the subset selection feature of tools like the TIGR Array Viewer would help identify more subtle discrepancies. Finally, Array Viewer allows the export of all or a subset of the data to flat files, databases, or other tools, such as Spotfire. Numerous other similar tools exist. Most

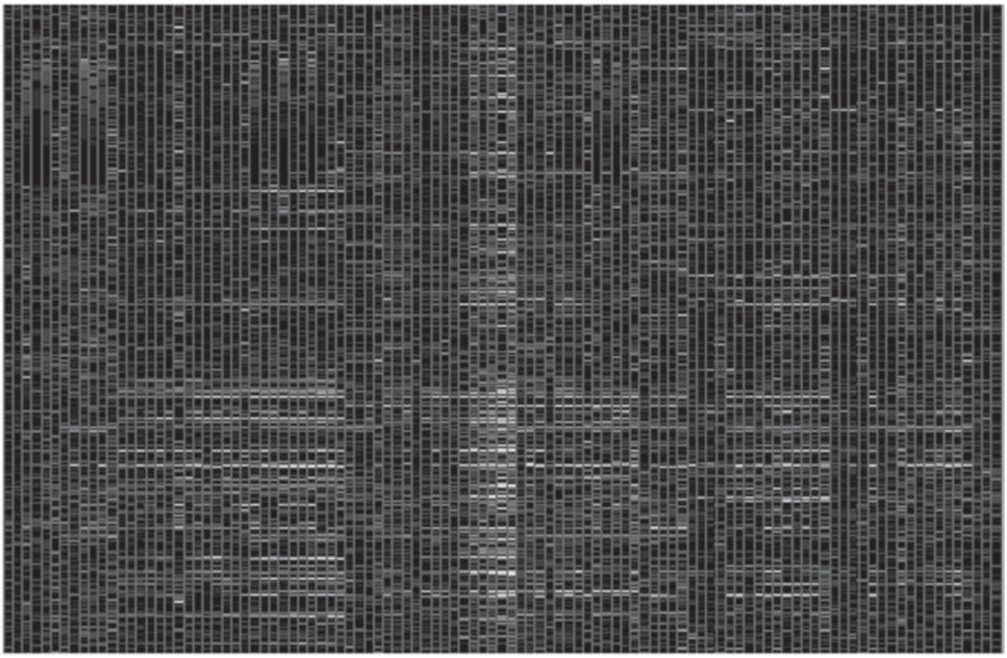


Fig. 13. Heat maps aid in the visualization of transcription profiling data. A heat map such as this one produced by Spotfire™, may be used to visually identify broad patterns of gene expression. Typically genes above and below user-defined thresholds are colored red (induction) and green (repression), respectively. Individual microarray experiments are organized on the X-axis, while individual genes are organized along the Y-axis. Several large groups of similar microarrays may be observed in this example. One might reasonably infer that similar gene expression patterns suggest similar mechanisms of action for the compounds used in these studies.

array platforms provide similar tools with at least some of the basic features provided by TIGR's Array Viewer. Many of these are proprietary and must be purchased, or are only available with purchase of that company's array products. Still others have been developed within pharmaceutical and biotechnology companies and are not generally available.

Data Analysis

Visualization of the data is a useful way to identify those genes that are most highly regulated. It is also a useful way to quickly discard uninteresting data, in order to generate a list of regulated genes. However, generating a list of regulated genes is only the first step in the analysis of TxP data and it is an inadequate use of microarray-derived experimental data. The purposes for running microarray experiments typically require advanced analysis of the TxP data. For example, simply asking what genes are expressed differently in an arthritic knee compared to a normal knee does little to help advance the effort to treat arthritis. Instead, identifying those genes whose expression correlates temporally and responds proportionally to the onset and severity of arthritic symptoms may help to identify a new target for intervention. Simply asking which genes are expressed in the liver in response to treatment with a pharmaceutical cholesterol lowering agent will do nothing to identify new targets for intervention

Table 2
Clustering Expression Data

ID	*Array 1*	*Array 2*	*ID*	*Array 1*	*Array2*
Gene 1	1200	700	Gene 9	630	1140
Gene 2	1050	750	Gene 10	600	470
Gene 3	990	730	Gene 11	540	140
Gene 4	700	500	Gene 12	420	480
Gene 5	690	1200	Gene 13	380	490
Gene 6	660	120	Gene 14	260	220
Gene 7	650	450	Gene 15	200	180
Gene 8	640	1090	Gene 16	190	250

into hypercholesterolemia. Identifying genes whose expression pattern clusters with the known targets of a cholesterol lowering drug may help identify mechanistic details of the body's cholesterol metabolism pathways. This could lead to the identification of new genes and gene products whose manipulation may also reduce dangerously high cholesterol levels. For these reasons, simply preparing a list of genes resulting from a microarray experiment falls short of the goal of understanding the science behind a particular signaling pathway of a disease mechanism.

Two general approaches were touched upon in the above paragraph, that of *clustering* and *correlation*. Both approaches can be applied to microarray data. Clustering expression data can help to identify co-regulated genes likely to be in common biochemical pathways. For example, proinflammatory agents such as tumor necrosis factor (TNF)-α, interleukin (IL)-1β, and lipopolysaccharide (LPS) are known to increase the expression of the cyclooxygenase (COX) 2 gene, which encodes a key enzyme involved in inflammation. The mechanism(s) by which these pro-inflammatory agents act is not completely understood. All of these agents are typically present outside the cell. Receptors on the surface of the cell detect their presence and initiate a signaling pathway that results in transcription factors inducing transcription of COX-2 mRNA. Even a carefully designed experiment including treatment with proinflammatory agents at several dose levels and/or time points would likely result in a large list of genes whose expression was affected. However clustering genes with similar expression patterns might identify novel genes with expression patterns similar to known members of the intracellular signaling pathway. Such novel genes may encode proteins that could be valuable points of intervention to block inflammation.

There are a number of approaches to clustering large sets of data. It would be difficult to describe each and new approaches are being developed regularly. One such common way to cluster microarray data is to use Euclidean distance in *n* dimensional space, where n is the number of variables for each microarray element. A very simple example will help visualize this approach. Imagine that two microarrays, each containing 16 elements that have been used to evaluate RNA from the livers of two rats treated with different carcinogenic compounds. The researcher might wish to cluster elements into groups of similar genes. Each element on the microarray has two variables, the signal from experiment or treatment 1, and the signal from experiment or treatment 2 (*see* Table 2). Simple visualization of the data may help to place the genes into several clusters, but it is not immediately clear how. However, graphing the data in n dimensional space, where n = 2 (for the number of variables for each element), will make clustering easier (*see* Fig. 14). Measuring a simple linear (Euclidean) distance between each point yields elements that are close to one another in this experimental space.

In this simplified example, one may not have been required. However, suppose a researcher performs TxP using a microarray with 10,000 elements on liver RNA from 12 rats fed 3 different diets resulting in different levels of fasted serum triglycerides. Using Euclidean distance, the researcher would graph each microarray element in 12-dimensional space. Clearly this must be done in virtual space using a computer, as the researcher cannot simply measure the linear distance between 10,000 points on a theoretical 12-dimensional graph. For this reason, methods have been developed to mathematically cluster the data points. One approach is to use a kth nearest neighbors strategy, where constant *k* equals a small integer. The computer program begins with a single data point, and finds it's 3 (if k = 3) nearest neighbors, that is, those points with the smallest Euclidean distance between them and the first, randomly selected element. Those 3 elements are the nearest neighbors of the first element, and they are all placed in the same *neighborhood*. The program then goes on finding the 3 nearest neighbors of each of the 3 neighbors it just added to the neighborhood. Each of these is then added into the neighborhood. The program does this iteratively until all of the nearest neighbors identified are already in that neighborhood. That neighborhood is a cluster. The program then goes on to another element not in the first neighborhood, and begins building the next neighborhood—the next cluster. In this example, the researcher could identify novel genes or known genes with no known function that are in the same cluster as genes known to be related to serum triglyceride levels.

There are several commercially available tools that enable the advanced analysis of microarray data, including clustering by several methods. As with microarray data, many visualization and reporting tools are proprietary and are not generally available. Increasingly, some of these tools are being made available to academic scientists. The TIGR Multi Experiment Viewer is a tool designed to facilitate the analysis of microarray data and is freely available to academics. It is a flexible and expandable tool capable of identifying patterns of gene expression and differentially expressed genes. TIGR Multi Experiment Viewer includes several data normalization, clustering and distance algorithms as well as a variety of graphical display features. Multi Experiment Viewer is fully compatible with TIGR's Array Viewer and represents a higher level of data visualization and analysis.

The European Bioinformatics Institute (EBI) also provides tools for genomic data analysis. The EBI Expression Profiler is a set of tools for the visualization and analysis of gene expression data. Their aim is to provide a complete and fully integrated set of analysis methods for expression data as well as sequence data. The value of including sequence data analysis capabilities with TxP data analysis tools lies in the fact that similar expression profiles of a set of genes may lead to the identification of sequence pattern profiles in the regulatory regions of these genes. Similarly, the relevance of shared sequence elements found in the promoters of a number of genes can be evaluated by comparing the expression profiles of those genes under multiple experimental conditions. Expression Profiler also allows for clustering of microarray data from multiple experiments (*see* Fig. 15). Genes whose expression patterns most closely match one another across many experiments cluster more closely than genes whose expression patterns are not closely matched. A dendogram, much like a phylogenetic tree is generated for each element, with branch points representing clusters.

See companion CD for color Fig. 15

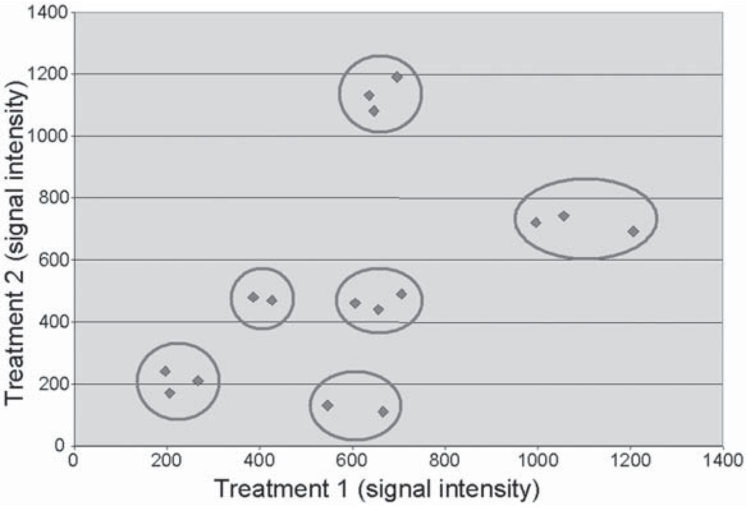


Fig. 14. Clustering of microarray data can be performed using a number of different techniques. Euclidean distance in n-dimensional space applies a variable derived from each microarray (a differential expression value or signal intensity) to every element present on the microarray. Each point on this graph represents an element on the microarray utilized. Elements in the same *neighborhood* are circled, representing six gene clusters.

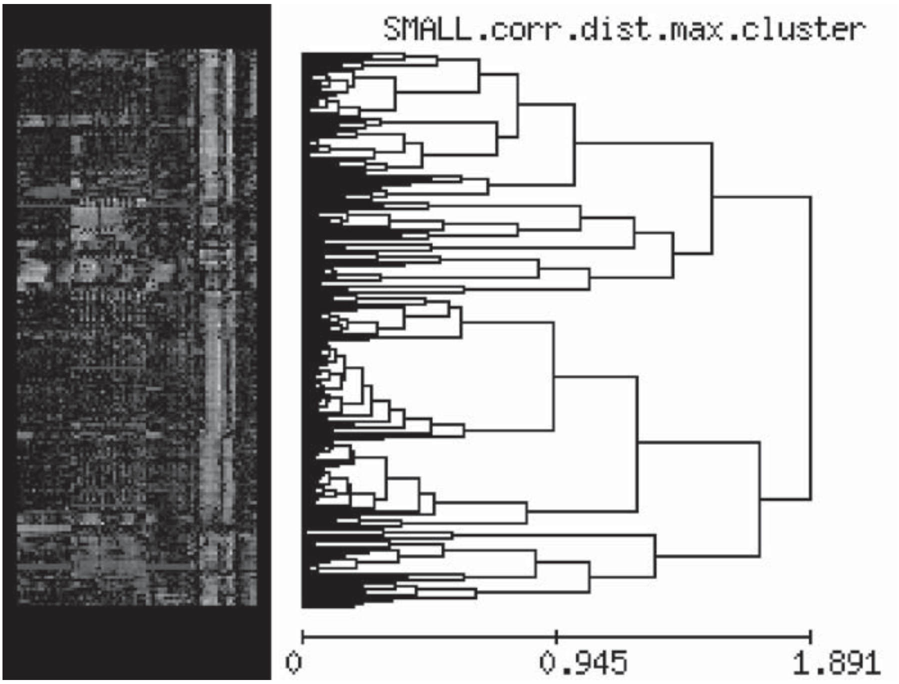
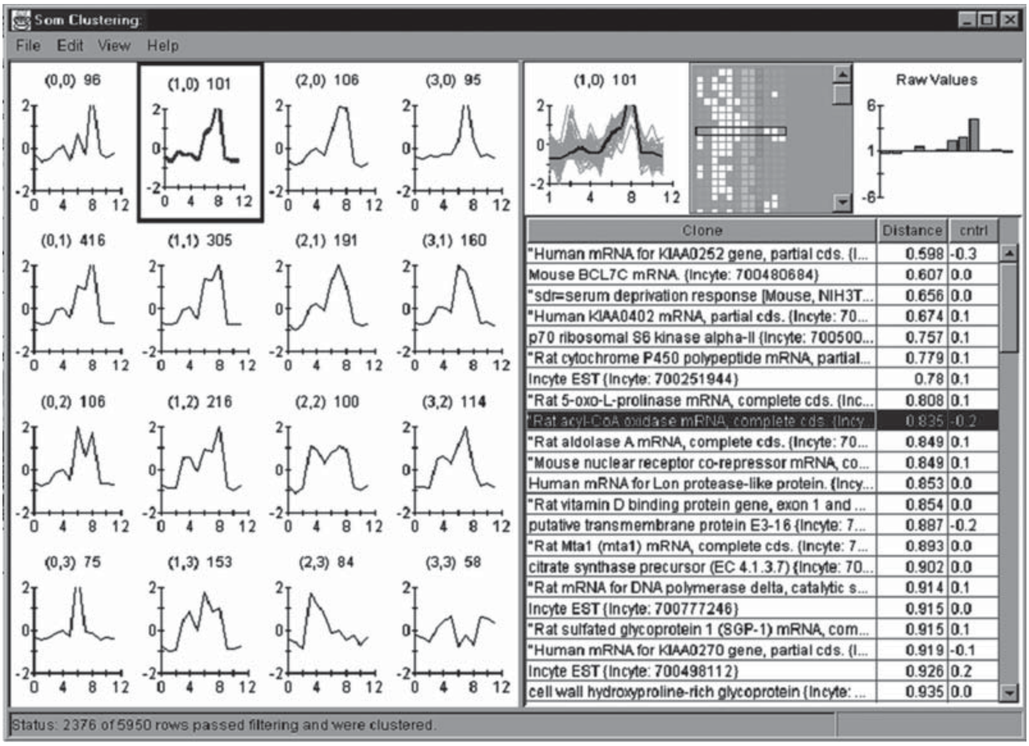


Fig. 15. The European Bioinformatics Institute's Expression Profiler tool allows users to cluster microarray data from multiple experiments. Genes whose expression patterns most closely match one another across many experiments cluster more closely than genes whose expression patterns are not closely matched. A dendogram is generated for each element, with branch points representing clusters.



Clone	Distance	cntrl
"Human mRNA for KIAA0252 gene, partial cds. (I...	0.598	-0.3
Mouse BCL7C mRNA. (Incyte: 700480684)	0.607	0.0
"sdr=serum deprivation response [Mouse, NIH3T...	0.656	0.0
"Human KIAA0402 mRNA, partial cds. (Incyte: 70...	0.674	0.1
p70 ribosomal S6 kinase alpha-II (Incyte: 700500...	0.757	0.1
"Rat cytochrome P450 polypeptide mRNA, partial...	0.779	0.1
Incyte EST (Incyte: 700251944)	0.78	0.1
"Rat 5-oxo-L-prolinase mRNA, complete cds. (Inc...	0.808	0.1
"Rat acyl-CoA oxidase mRNA, complete cds. (Incy...	0.835	-0.2
"Rat aldolase A mRNA, complete cds. (Incyte: 70...	0.849	0.1
"Mouse nuclear receptor co-repressor mRNA, co...	0.849	0.1
Human mRNA for Lon protease-like protein. (Incy...	0.853	0.0
"Rat vitamin D binding protein gene, exon 1 and ...	0.854	0.0
putative transmembrane protein E3-16 (Incyte: 7...	0.887	-0.2
"Rat Mta1 (mta1) mRNA, complete cds. (Incyte: 7...	0.893	0.0
citrate synthase precursor (EC 4.1.3.7) (Incyte: 70...	0.902	0.0
"Rat mRNA for DNA polymerase delta, catalytic s...	0.914	0.1
Incyte EST (Incyte: 700777246)	0.915	0.0
"Rat sulfated glycoprotein 1 (SGP-1) mRNA, com...	0.915	0.1
"Human mRNA for KIAA0270 gene, partial cds. (I...	0.919	-0.1
Incyte EST (Incyte: 700498112)	0.926	0.2
cell wall hydroxyproline-rich glycoprotein (Incyte: ...	0.935	0.0

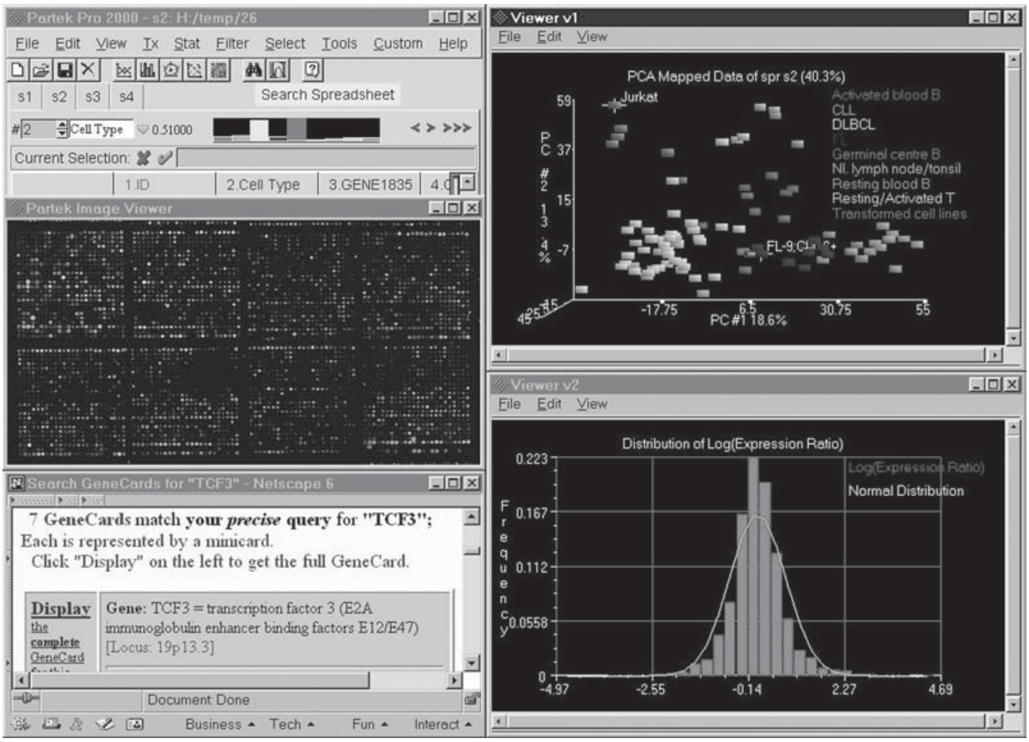
Status: 2376 of 5950 rows passed filtering and were clustered.

Fig. 16. Self-organizing maps represent to cluster microarray data into groups of similarly regulated genes. Millennium Pharmaceuticals and others provide tools that enable scientists to generate self-organizing maps from their TxP data.

Another clustering approach that has been taken is the self-organizing map (SOM). Self-organizing maps were developed by Teuvo Kohonen (1995) as a way to visualize high-dimensional data. The self-organizing map is an unsupervised learning algorithm that attempts to reduce the dimensions of data using self-organizing neural networks. The SOM algorithm finds vectors representative of the input data set. At the same time the algorithm creates a continuous map from the input space mapped onto a 2-dimensional lattice. The preliminary vectors of the SOM are initialized to random values. With iteration of each training pattern the winning *neurone* is found by comparing input (data) vectors and weight vectors of the neurones using Euclidean distance. The weights of the winning neurone and its nearest neighbors are moved toward the input vector using a learning rate factor. The SOM can also be used as a starting point for additional analyses. Figure 16 shows an example of a self-organizing map output generated by Millennium Pharmaceutical's SOM tool. The data was divided into four groups of four patterns. The highlighted cluster contains 101 genes, one of which, acyl CoA oxidase, is highlighted in the right panel. The individual expression pattern for this gene is overlaid upon the expression patterns of the other members of the group in the upper middle panel (*see* Fig. 16).

See companion CD for color Fig. 16

Higher order statistical analysis of transcription profiling data is another valuable means to sort through the thousands of data points that result from TxP experiments.



7 GeneCards match your ***precise*** query for "TCF3";
Each is represented by a minicard.
Click "Display" on the left to get the full GeneCard.

Display the **complete** GeneCard

Gene: TCF3 = transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
[Locus: 19p13.3]

Fig. 17. Partek Pro 2000 allows scientists multiple data analysis options. Principal component analysis, multidimensional scaling, and inferential analysis can all be performed by tools such as Partek Pro. These higher-order statistical analyses of microarray results are vital to avoid false-positives and false-negatives. The costs of following up on false results can be staggering. As transcription profiling finds application in medical diagnostics, false results may even endanger patients.

WWW

Partek Pro 2000 (*see* Website: http://www.partek.com/) is a comprehensive data visualization and data-analysis tool for performing statistical analyses on many types of data. Partek Pro 2000 is not limited to TxP data, but can also be used for high throughput screening data, proteomic data, and nonbiological data. The software can be adapted to work with numerous databases and data formats and has been used by many large pharmaceutical, biotechnology, and bioinformatics companies. Partek Pro includes modules that permit high-level statistical analysis such as principal component analysis, multidimensional scaling, and inferential analysis (*see* Fig. 17). The tool can be used to apply statistical rigor to experimental design, normalization, chip-to-chip scaling, and other activities performed during the analysis. One difficulty in applying statistics to TxP data lies in the small sample size. That is, due to financial and biological constraints, it is not uncommon to have only a very small number of samples subjected to microarray analysis. Statisticians typically require high *n*, that is a high number of experimental repetitions to determine statistically significant values with any precision. If an experiment includes only three animals per treatment group, only very large differential expression values may be statistically significant. Addi-

See companion CD for color Fig. 17

tionally, comparing 10,000 elements with only a small number of biological samples will yield many seemingly significant relationships due to chance alone. Partek Pro 2000 and similar tools can be used to apply rigorous statistical tests to determine whether seemingly relevant relationships between two or more elements might be occurring simply by chance.

As well as clustering genes by similar expression patterns, correlating the expression of genes with one or more relevant phenotypes is another way to identify important and interesting genes. This approach may be particularly useful for pharmacogenomic studies, in which one or more genes are sought with a causative role in a specific measurable response or phenotype within a large population. For example, if a small but significant population does not respond to a particular pharmaceutical agent, transcription profiling may identify a gene or pathway whose regulation is different in the unresponsive population than in the responsive population. When studying thousands of genes at once using microarrays, such a simple pattern of expression would be very difficult to identify by simple visual inspection. Furthermore, clustering may not help, as known reference genes may not be available. One approach to this issue would be to express the phenotypes as a balanced differential expression, then cluster this value phenotype with the entire microarray dataset. In this way, identification of those genes that occur in the same cluster as the phenotype of interest ought to display expression patterns that correlate to the phenotype.

Another approach is to calculate a correlation score based upon a number of parameters. A simple correlation coefficient can be generated for each microarray element compared to one or more phenotypic values. This score may be adjusted to reflect signal intensities and significant differential expression values, to weight the scores in favor of reliable genes. All of these clustering and correlation approaches assume that all of the differential expression values are genuine, even those within the noise (*see* Fig. 10B). By weighting signal intensity, the correlation score approach is less likely to identify genes whose expression lies within the more nebulous range of differential expression values. Such an approach will identify genes that are both highly regulated in at least some of the experimental conditions, and correlative with the phenotype of interest. It is important to note that, just because an element has a small BDE that occurs within the noisy region of the scatter plot, does not mean that the value is not an accurate reflection of the expression of the gene that the element represents. However, owing to the general lack of a large number of replicates, TxP results should be validated by alternative methods, such as RT-PCR or the Quantigene™ (Bayer Diagnostics) assay. Genes with large changes in their expression are more reliably and more easily verified by secondary methods. Additionally, genes with very small changes in expression will be difficult to measure and verify without significant experimental repetition, making them poor choices for markers of efficacy. Since the purpose of TxP experiments is often the identification of molecular markers of disease or efficacy, or the identification of new targets for intervention, tracking changes in the expression of the genes identified will be key to additional experiments.

With these caveats in mind, the ability to identify genes whose expression correlates with a phenotype of interest, and whose expression under at least some of the experimental conditions is highly regulated is a valuable way to identify interesting genes. In a theoretical experiment described earlier, TxP was performed using liver RNA from 12 rats fed three different diets. The dietary regimens resulted in differ-

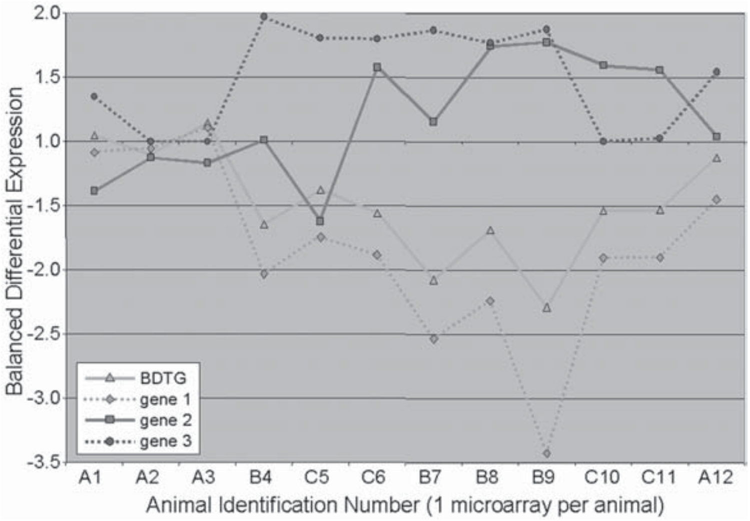


Fig. 18. Correlation analysis can be used to identify genes whose expression correlates with a specific phenotype of interest. In the experiment shown, 12 rats were randomly assigned to one of three diet groups, A, B, or C for 4 wk. Fasted serum triglycerides were measured in individual rats prior to sacrifice. Liver RNA was used to prepare probes for hybridization onto microarrays. The triglyceride phenotype was used to calculate a balanced differential triglyceride (BDTG) value, using the same equation to determine differential expression values for the array elements. The BDTG value was then compared to each element. In this example, the expression of gene 1 correlates well with the measured phenotype.

ences in fasted serum triglycerides relative to one another. A researcher knowing nothing about gene products involved in the dietary regulation of serum triglycerides might wish to identify genes whose expression is directly proportional to this phenotype, which is a risk factor in humans for cardiovascular disease. Figure 18 shows a graphical representation of three genes and the serum triglyceride phenotype from such an experiment. The serum triglycerides were first expressed as a Balanced Differential TriGlyceride (BDTG) value, using Equation 1 detailed earlier. Instead of a treated and control, the average TG value for the four rats fed the control diet (diet group A) was used as the control *signal intensity*. Individual TG values were used as the treated *signal intensities*, and compared to the pooled control, much as individual rat liver RNA samples would be compared to a pooled control sample on the microarray. A correlation score could be calculated for each element compared to one or more phenotype. High scoring genes, such as gene 1 in Fig. 18, would correlate well with the phenotype. That is, those animals with the lowest serum triglyceride levels relative to control have the largest decreases in the expression of gene 1 relative to the control group. Poorly scoring genes, such as genes 2 and 3 (*see* Fig. 18) correlate poorly with the phenotype. Without the ability to rapidly generate correlation scores for each element on the array, it would have been necessary to graph each element along with the phenotype until those with similar graphs were identified. Clearly this is not feasible when performing experiments using thousands of arrayed elements. Software that automates this approach would enable a researcher to rapidly identify genes with *interesting* expression patterns.

Project Organization

The final type of informatic task required to maximize the value of microarray experiments is project organization. Every microarray has many features, many pieces of associated data that must be stored, linked, and made available to the scientists performing and analyzing the experimental results. There are a number of organizations currently building data repositories for microarray experimental results. The National Center for Biotechnology Information (NCBI) has developed a gene expression data repository and online resource for the retrieval of gene expression data called the gene expression omnibus (*see* Website: http://www.ncbi.nlm.nih.gov/geo/). The gene expression omnibus supports multiple types of expression data, including microarray data and SAGE (Serial Analysis of Gene Expression) data. Additionally, the International Life Sciences Institute (ILSI) (*see* Website: http://www.ILSI.org/) in collaboration with the European Bioinformatics Institute (EBI) has developed both a database and a standardized reporting format for microarray data. Among the challenges facing such efforts are ways to ensure that minimal data quality standards are met and the need to include sufficient experimental information, perhaps including actual .tiff files, without taxing servers to the point of making data downloads prohibitive.

Databases that simply store microarray data are a valuable way of ensuring public access to a significant body of microarray data. However, many organizations are developing database and analysis software suites that allow for interactive data analysis on all or a subset of data stored in a database. Links are also included to associated histopathological images from every tissue for which profiling data has been acquired. The Phase-1 product is aimed toward the field of toxicogenomics, the use of genomic and microarray technology in toxicology and preclinical safety assessment, but their approach to an interactive data repository has clear advantages over a simple data repository from which information must be exported prior to data analysis.

Another application of informatics to the field of genomics that has not been fully appreciated by the scientific community is the need for rational, statistically informed experiment planning. There are many types of microarrays. A complete statistical analysis of the performance of the type of array being used in a laboratory is vital. For example, knowing the sensitivity and reproducibility of profiling results on the array platform being used can inform decisions regarding the number of samples required, and the reliability of the results of each array experiment are. An array platform that gives consistently noisy results may require large groups or many duplicates to provide useful data. Alternatively, if only small amounts of biological material are available for microarray analysis, the confidence limits may require that only genes with very large differential expression values are reliable. In such cases, correlations with phenotypic parameters would be unreliable. These issues date back to quality control issues. At this time, this author is unaware of any tool that seeks to address this issue. However tools such as Partek Pro 2000 can allow the higher-level statistical analysis necessary to evaluate the experimental parameters of TxP experiments and direct planning of future experiments. There are several books that detail statistical approaches to experimental design. One such book is *Biostatistics and Experimental Design,* by James Zolman (1993). The area of statistical experimental design tends to be overlooked by many biologists, to the detriment of their experiments. With the large number of elements that can be arrayed, the probability of relationships that may seem significant occurring simply by chance is very large. The lack of rigorous statistical evaluation of study designs *prior to* the inception of the studies will result in a significant loss of time and money trying to validate questionable TxP results.

Glossary and Abbreviations

BDE Balanced differential expression. A way of expressing fold changes in expression.

Bioinformatics A field of study encompassing statistical and computational approaches to gathering, archiving, annotating, visualizing, analyzing and reporting genomic data.

cDNA Complementary DNA. Generated by reverse transcription of RNA.

Cy3 Cyanine 3 dye. A fluorescent dye used to label cDNA or cRNA populations.

Cy5 Cyanine 5 dye. A fluorescent dye used to label cDNA or cRNA populations.

fBDE Fractional increase balanced differential expression. A way of expressing changes in gene expression.

Genomics An emerging scientific field encompassing the large scale study of the informational content and regulatory mechanisms of entire genomes.

mRNA Messenger RNA. Typically encodes proteins, often regulated by environmental conditions.

PCA Principal components analysis. A method of clustering complex data sets.

Proteomics The study of the content and regulatory mechanisms the entire complement of proteins encoded by an organism's genome.

RT-PCR Reverse transcriptase polymerase chain reaction. A method of evaluating changes in gene expression.

SOM Self-organizing map. A method of clustering complex data sets.

Transcription Profiling A technique whereby changes in the expression of many genes are measured in a single experiment.

Suggested Readings

Kohonen, T. (1995) Self-Organizing Maps, Springer, NY.

Zolman, J. (1993) Biostatistics and Experimental Design, Oxford University Press, New York, NY.

35 Microarrays

Tools for Gene Expression Analysis

Sorin Draghici

Introduction

A particular tool that has been shown to be very useful in genomics is the DNA microarray. In its most general form, the DNA array is a substrate (nylon membrane, glass, or plastic) on which DNA is deposited in localized regions arranged in a regular, grid-like pattern. Two main approaches are used for microarray fabrication: *in situ* synthesis and deposition of DNA fragments. *In situ* manufacturing can be further divided into photolithography, ink jet printing, and electrochemical synthesis. The photolithographic approach (Affymetrix) is similar to the VLSI fabrication process: photolithographic masks are used for each base. If a region should have a given base, the corresponding mask will have a hole allowing the base to be deposited at that location. Subsequent masks will construct the sequences base by base. This technology allows the fabrication of very high-density arrays but the length of the DNA sequences constructed is limited[1]. The ink jet technology (e.g., Agilent and Protogene) is similar to the technology used in ink jet color printers. Four cartridges are loaded with the A, C, G, and T nucleotides. As the print head moves across the array substrate, specific nucleotides are deposited where they are needed. The electrochemical synthesis (CombiMatrix) uses small electrodes embedded into the substrate to manage individual reaction sites. Solutions containing specific bases are washed over the surface and the electrodes are activated in the necessary positions. In the deposition-based fabrication (e.g., Clontech and Corning), the DNA is prepared away from the chip. Robots dip thin pins in the solutions containing the desired DNA and then touch the pins onto the surface of the arrays. Small quantities of DNA are deposited as spots on the array. Unlike *in situ* manufacturing, spotted arrays can use small sequences, whole genes, or polymerase chain reaction (PCR) products.

The DNA array is subsequently probed with complementary DNA (cDNA) obtained by reverse transcription of mRNA extracted from a tissue sample. This DNA is fluorescently labeled with a dye and subsequent illumination with an appropriate

[1] This is because the probability of introducing an error at each step is very small but not zero. In order to limit the overall probability of an error, one needs to limit the length of the sequences. To compensate, many short sequences from the same gene can be synthesized on a given array. The particular sequences must be carefully chosen to avoid cross-hybridization.

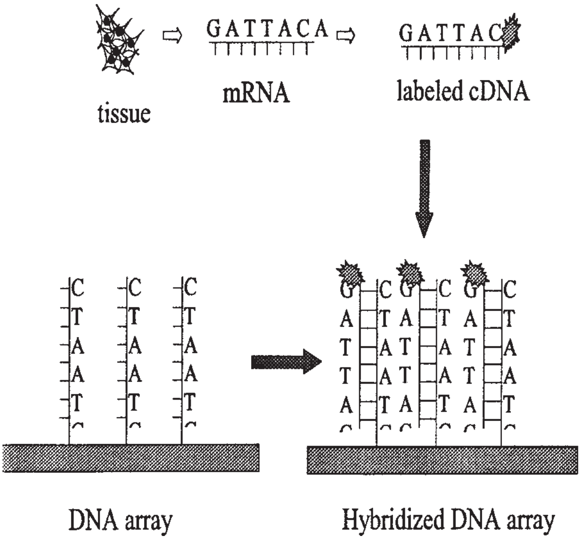


Fig. 1. A general overview of the DNA array use. The mRNA extracted from tissue is transformed into cDNA which is hybridized with the DNA previously spotted on the array.

light source will provide an image of the array. The intensity of each spot or the average difference between matches and mismatches can be related to the amount of mRNA present in the tissue which in turn is usually correlated with the amount of protein produced by the gene corresponding to the given region. The labeling can also be done with a radioactive substance. Figure 1 illustrates the microarray process. In a multichannel experiment, several probes are labeled with different dyes and used at the same time in a competitive hybridization process. A large number of expression values are obtained after processing the image(s) of the hybridized array. Typically, one DNA chip will provide expression values for thousands of genes.

Although microarrays have been used successfully in a range of applications including sequencing and single nucleotide polymorphism (SNP) detection, most applications are related to gene expression. Typical examples include comparing healthy and malignant tissue, studying cell phenomena over time or studying the effect of various factors on the global pattern of gene expression.

Challenges

Compared to other molecular biology techniques, microarrays are relatively new and a number of issues and challenges remain. Such issues include:

Noise

Due to their nature, microarrays tend to be very noisy. Even if an experiment is performed twice with exactly the same materials and under exactly the same conditions, it is likely that after the scanning and image-processing steps, many of the same genes will probably be characterized by different quantification values. Noise is introduced at each step of various procedures[2]: mRNA preparation (tissues, kits,

[2] Not all steps apply to all types of arrays.

and procedures vary), transcription (inherent variation in the reaction, enzymes), labeling (type and age of label), amplification, pin type (quill, ring, ink-jet), surface chemistry, humidity, target volume (fluctuates even for the same pin), slide imperfections (slide production), target fixation, hybridization parameters (e.g., time, temperature, buffering), nonspecific hybridization (labeled cDNAs hybridized to areas that do not contain perfectly complementary sequences), nonspecific background hybridization (e.g., bleeding with radioactive materials), artifacts (dust), scanning (gain settings, dynamic range limitations, inter-channel alignment), segmentation (feature/background separation), and quantification (mean, median, percentile of the pixels in one spot).

The challenge appears when comparing different tissues or different experiments. Is the variation of a particular gene due to the noise or is it a genuine difference between the different conditions tested? Furthermore, when looking at a specific gene, how much of the measured variance is due to the gene regulation and how much to noise? The noise is an inescapable phenomenon and the only weapon that the researcher seems to have against it is replication.

Normalization

The aim of the normalization is to account for systematic differences across different data sets, e.g., overall intensity and eliminate artifacts, e.g., nonlinear dye effects. Normalization is crucial if results of different experimental techniques are to be combined. While there is a general consensus that normalization is required, there is little if any such consensus regarding how normalization should be accomplished. Normalization can be necessary for different reasons such as different quantities of mRNA (leading to different mean intensities), dye nonlinearity, and saturation toward the extremities of the range.

Experimental Design

The experimental design is a crucial but often neglected phase in microarray experiments. If the experiments are not adequately designed, no analysis method will be able to obtain valid conclusions. It is very important to provide data for a proper comparison for every major source of variation. In a classical example from agriculture, one wishes to compare two strains of corn. If each strain is planted on a different field, there will be no way to distinguish the effect of the field including more nutrients or more sun, from the effect of the corn strain. This is a typical example of a confounding experimental design. A better design would have planted opposite halves of each field with both strains of corn such that an analysis of the results would allow a distinction between the two effects. Experimental design in the microarray context has been thoroughly explored.

Large Number of Genes

The fact that microarrays can interrogate thousands of genes in parallel is one of the features that led to the wide adoption of this technology. However, this characteristic is also a challenge. The classical metaphor of the needle in the haystack can easily become an accurate description of the task at hand when tens of thousands of genes are investigated. Furthermore, the sheer number of genes can change the quality of the phenomenon and the methods that need to be used. The classical example is that of the p-values in a multiple testing situation.

Significance

If microarrays are used to characterize specific conditions, a crucial question is whether the expression profiles differ in a significant way between the groups considered. The classical statistical techniques that were designed to answer such questions, e.g., chi-square tests, cannot be directly applied because the number of variables, usually thousands of genes, is much greater than the number of experiments, usually tens of experiments.

Biological Factors

In the normal cell, the RNA polymerase transcribes the DNA into mRNA which carries the information to ribosomes where the protein is assembled by tRNA during translation. Most microarrays measure the amount of mRNA specific to particular genes and the expression level of the gene is associated directly with the amount of mRNA. However, the real expression of the gene is the amount of protein produced not the amount of mRNA. Although in most cases, the amount of mRNA accurately reflects the amount of protein, there are situations in which this may not be true. If nothing else, this is a fundamental reason why microarray results usually require verification. Microarrays are tools for screening many genes and focusing hypothesis. However, conclusions obtained with microarrays must be validated with independent assays using different techniques from various perspectives.

Array Quality Assessment

It is useful if data analysis is not seen as the last step in a linear process of microarray exploration but rather as a step that completes a loop and provides the feedback necessary to fine tune the laboratory procedures that produced the microarray. Thus, array quality assessment is an aspect that should be included among the goals of the data analysis. It would be very useful if in addition to the expression values the analysis provides some quality assessment of the arrays used. Such quality measures will discard the data below a standard as well as the identity of possible causes of failure in the process.

Current Issues in Microarray Data Analysis

Normalization

We assume that each spot is represented by a value computed from the pixel intensities. A first pre-processing step is the background correction. This can be done locally[3], using a group of spots (e.g., a subgrid)[4] or blank spot[5]. Once the background correction has been applied, a usual step in two-channel experiments is to consider the ratio between the two channels (e.g., experiment/reference). Subsequently, one should apply a logarithmic function (log). There are several reasons for this. First, the log makes the distribution symmetrical. Consider the values 100,

[3] Only the background local to the spot is considered; good when the background varies considerably from spot to spot.

[4] Suitable for high density arrays where there are not enough pixels around a single spot to produce a reliable value.

[5] Some researchers have noted that a spot containing DNA that is not supposed to hybridize will often have an intensity lower than the background suggesting that subtracting the background might overcorrect the values. In order to control this more accurately, one can use spots containing no DNA or spots containing foreign DNA that should not hybridize.

1000, and 10,000 (a 16-bit tiff file can contain values between 1 and 65,536). If one considers the difference (distance) between the middle values and the two extremes, one is tempted to consider the difference to the right more important than the difference to the left (10,000 – 1000 = 9000 >> 1000 – 100 = 900). However, from a biological point of view the phenomenon is the same, namely there is a ten fold change in both cases. The log will transform the values into $log(100)$, $log(1000)$, and $log(10{,}000)$, which for log base 10 are 2, 3, and 4, rejecting the fact that the phenomena are the same only that they happen in different directions. Second, if the log is taken in base 2, subsequent analysis is facilitated, e.g., selecting genes with a four-fold variation can be done by cutting a ratio histogram at the value $log(ratio) = 2$. Furthermore, the log partially decouples the variance and the mean intensity and makes the distribution almost normal. Replicate genes can be combined by calculating mean, median, or mode and coefficients of variance (cv). This will produce the average-log-ratios.

One important problem is that various dyes produce different results due to different biochemical properties. The idea of a flip dye experiment was introduced in order to control such phenomena. In a flip dye experiment, the two samples of mRNA, A and B are labeled first with cy3 (A) and cy5 (B) and then with cy5 (A) and cy3 (B). Subsequent hybridization and image analysis will produce two sets of data that represent the same biological sample. A plot of the expression levels registered on the two channels for the same mRNA should produce a straight line $cy3 = cy5$. One is prepared to accept a small number of genes that are off the straight line of reference but any general trend suggests a non-negligible influence that should be corrected. Similar expectations even hold for experiments involving all or most genes in a genome. It is assumed that most genes will not change. Such expectations are not warranted if a small subset of genes is considered, especially if the genes are known to be functionally related.

The final normalization step tries to obtain values that are independent of the experiment and hopefully can be compared with other values. This can be carried out in different alternative ways as follows.

Method 1

Divide the values of each channel/experiment by their mean[6]. This corrects global intensity problems, such as one channel being more intense than the other, but does not address nonlinearity of the dye. Variations of this strategy eliminate the values in the upper and lower 10% of the log ratio distribution (corresponds to differentially regulated genes) divided by the mean of the remaining genes. The rationale being that few genes will be differentially regulated but most genes will remain the same. Eliminating the tails of the log ratio distribution theoretically eliminates the differentially regulated genes and dividing by the mean of the remaining genes bring the two sets of measurements to a common denominator.

Method 2

Modify the values of each channel/experiment such that certain control spots in both experiments have the same or similar values. The control spots should span the entire intensity range.

[6] The mean can be substituted with the median, mode or percentile if the distribution is noisy or skewed. One could also subtract the mean and divide by the standard deviation.

Method 3

Apply iterative linear regression as follows: apply linear regression, find residuals, eliminate spots that have residuals more than 2 SD away from zero (potential differentially regulated genes), apply regression and continue until the residuals become less than a given threshold.

Method 4

Apply a parametric nonlinear normalization with or without the previous division by mean, median or percentile. The exponential normalization fits an exponential decay of the form: $y = a + b \cdot e^{-cx}$ to the $log(Cy3/Cy5)$ vs $log(Cy5)$ curve[7]. Subsequently, the values are normalized by subtracting the fitted log ratio from the observed log ratio. This normalization essentially relies on the hypothesis that the dye effect can be described by an exponential curve, which may not be true for other dyes. Furthermore, the fitting of the exponential is computationally intensive involving a least-mean-square (LMS) minimization.

Method 5

Apply a general nonlinear normalization such as Loess: the values are modified in order to obtain a linear variation of the two channels (in a flip dye experiment). This is usually computationally intensive. An excellent correction for the dye nonlinearity without the high computational cost of Loess can be obtained using an adaptive piecewise linearization method.

Figure 2 illustrates the importance of pre-processing and normalizing the cDNA data labeled with cy3/cy5. The array contains 944 genes spotted in triplicates. The raw data is present in Fig. 2A. Owing to the different properties of the two dyes, the intensities measured on one channel are consistently higher than the intensities measured on the other channel. In these conditions, it is likely that any analysis method will yield erroneous results. Taking the logarithm helps greatly, as illustrated in Fig. 2B, but the overall intensity difference between the two channels is still present. Dividing by the mean brings most points in the vicinity of the reference line finally aligning the data with the reasonable expectation that most genes will not change between the two experiments. The steps taken here would be an example of the very minimum preprocessing necessary for two-channel cDNA data. Note that there is no correction for dye nonlinearity, which is rejected in the curvature of the data points distribution. Correcting this nonlinearity would further enhance the accuracy of any subsequent analysis.

It is important to note that the measured variance is dependent on the mean intensity having high variance at low intensity levels and low variance at high intensities. This can be corrected by using an iterative algorithm that gradually adjusts the parameters of a probabilistic model. This approach can be further refined using a Gamma-Gamma-Bernoulli model.

Normalization is slightly different for oligonucleotide arrays (e.g., Affymetrix's GeneChips) designed for gene expression analysis. Here, a gene is represented by a number of probe pairs (short oligonucleotide sequences) with each pair containing a perfect match and a mismatch (the same sequence but for a different nucleotide). The

[7] The reference line in a $log(cy3/cy5)$ vs $log(cy5)$ plot is $y = 0$ because $log(cy3/cy5) = log(cy3) - log(cy5) = 0$ if cy3 equal to cy5, as expected for the same mRNA labeled differently.

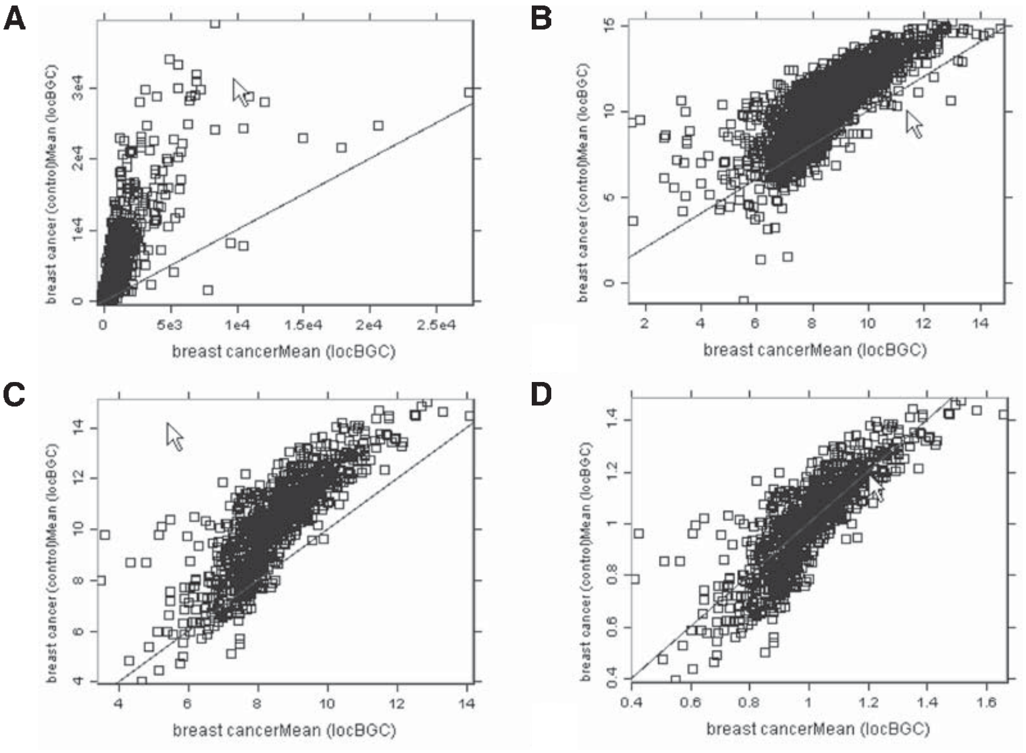


Fig. 2. The effects of data normalization. The data represents a *cy*3/*cy*5 cDNA array with 944 genes printed in triplicates. **(A)** Raw data; the data is far from the reference and the scales are different on the two axes. **(B)** After log; the data is closer to the reference line but still above it. **(C)** Replicates are combined; each gene is represented as the mean of its replicates. **(D)** Subtracting the mean finally centers the data; however, the data are still curved reflecting the nonlinearity of the dyes.

amount of hybridization on the mismatched sequence is thought to be representative of non-specific hybridization and its intensity is usually subtracted from the intensity of the perfect match. Subsequently, such differences are averaged for all probes corresponding to a gene and an average difference (AD) is calculated. The higher this AD, the higher the expression level of the gene. The software provided by Affymetrix also calculates a *call*, i.e., a ternary decision about the gene: absent (A), marginally present (M), and present (P). A conundrum can occur if absent genes have average difference higher than present genes. Options include ignoring the calls and using only the average differences of all genes, considering only genes that are declared present, ignoring the calls and considering only genes with the average difference larger than a threshold or calculating the expression values according to some other model.

Selection of Differentially Regulated Genes and Significance Analysis

The measurements obtained from microarray experiments have multiple sources of variation. There is approx a 5% probability that the hybridization of any single spot containing complementary DNA will not reject the presence of the mRNA and a 10% probability that a single spot will provide a signal even if the mRNA is not present.

Given this, it is crucial to distinguish between interesting variation caused by differential gene expression and variation introduced by other causes. Using replicates at any or all levels (e.g., spots, arrays, mRNA preparations) is the only known method that allows subsequent discrimination.

Selecting interesting genes can be done in various ways. Two selection methods are widely used. The first selection method is the *fold method*. This method involves a simple comparison between the expression levels in the experiment and the control. Genes having different values in experiment vs control will be selected. Typically, a difference is considered significant if it is at least two- or three-fold.

The second widely used selection method is the *ratio method*. This method involves selecting the genes for which the experiment/control ratio is a certain distance, e.g., 2 SD, from the mean experiment/control ratio. Other ad hoc thresholding procedures have also been used. For example, one can select genes with at least a two-fold change and for which the difference between the duplicate measurements does not exceed half their average. These criteria can be expressed in terms of a ratio mean and SD and therefore this is a variant of the ratio method. An alternative method selects the genes for which the absolute difference in the average expression intensities is much larger than the estimated standard error.

More complex methods such as analysis of variance (ANOVA), maximum likelihood approach, gene shaving, assigning gene confidence or significance, bootstrap, and Bayesian approaches have also been proposed but are not widely used. Other interesting methods take into consideration that the variance depends on the intensity. Intensity dependency can be reduced to defining some curves in the green-red plane corresponding to the two channels. The points inside these curves correspond to genes that change little between the control and experiment while the points that fall outside the curves correspond to differentially regulated genes.

If several arrays are available for each condition, differentially expressed genes can be identified by ranking the genes according to their similarity to an expected expression profile. False-positives can be controlled through random permutations that allow the computation of suitable cut-off thresholds.

Another possible approach to gene selection is to use univariate statistical tests (e.g., t-test) to select differentially expressed genes. Regardless of the particular test used (e.g., t-test if normality is assumed), one needs to consider the fact that when many genes are analyzed at one time, some genes will appear as being significantly different just by chance.

Let us assume we are considering a gene with a value (e.g., log-ratio) v situated in the tail of the histogram of all such values, possibly indicating that the gene is regulated. The p value provided by the univariate test is the probability that v is where it is just by chance. If we call this gene differentially regulated based on this value and the value is there by chance, we will be making a mistake. Therefore, p is *the probability of making a mistake*. The probability of drawing the right conclusion in this one experiment will be $1 - p$. If there are R such experiments, we would like to draw the right conclusion from *all* of them. The probability of this will be $prob(right) = (1 - p)^R$. The probability of making a mistake will be $prob(wrong) = 1 - prob(right) = 1 - (1 - p)^R$. This is called *Sidák correction*. Bonferroni noted that for small p, $1 - (1 - p)^R \quad Rp$ and proposed $\tilde{p} = p/R$. Both Bonferroni and Sidák corrections are not suitable for gene expression analysis because for large number of genes R, no gene will be below the corrected p value (e.g., $\tilde{p} = p/R$ for Bonferroni).

A family of methods that allow less conservative adjustments of the p-values without the heavy computation involved in resampling is the Holm step-down group of methods. These methods order the genes by increasing order of their p-value and make successive smaller adjustments.

Bonferroni, Siák and Holm's step-down adjustment assumes that the variables are independent. This may not be true for expression data since genes influence each other in complex interactions. The Westfall and Young (W-Y) step-down is a more general method which adjusts the p-value while taking into consideration the possible correlations. Duplication, together with a univariate testing procedure (e.g., t-test or Wilcoxon) followed by a W-Y adjustment for multiple testing are proposed.

Another technique that considers the correlation is the bootstrap method. This method samples with replacement the pool of observations to create new data sets and calculates p-values for all tests. For each data set, the minimum p-value on the resampled data sets is compared with the p-value on the original test. The adjusted p-value will be the proportion of resampled data where the minimum pseudo-p-value is less than or equal to an actual p-value. Bootstrap used with sampling without replacement is known as the permutation method. Both bootstrap and permutation are computationally intensive.

Data Analysis Techniques

Scatterplots

The scatterplot is probably the simplest tool that can be used to analyze expression levels. In a scatterplot, each axis corresponds to an experiment and each expression value of a given gene is represented as a point. If a gene G has an expression level of e_1 in the first experiment and that of e_2 in the second experiment, the point representing G will be plotted at coordinates (e_1, e_2) in the scatterplot (*see* Figs. 2 and 3). In such a plot, genes with similar expression levels will appear somewhere near the line ($y = x$). The further from the diagonal, the more significant the variation between experiments.

The main disadvantage of scatterplots is the fact that they can only be applied to a very small number of dimensions since they can only be plotted in two or three dimensions. Dimensionality reduction techniques such as Principal Component Analysis are generally used to extend the usefulness of scatterplots.

Principal Component Analysis

One very common difficulty in many problems is the large number of dimensions. A natural approach is to try to reduce the number of dimensions and thus, the complexity of the problem, by eliminating those dimensions that are not "important." Of course, the problem now shifts to defining what is an important dimension. A common statistical approach is to pay attention to those dimensions that account for a large variance in the data and to ignore the dimensions in which the data does not vary much.

This is the approach used by Principal Component Analysis (PCA). The PCA approach is shown in Fig. 4. The data that includes patterns from two classes, red crosses and blue circles, is given in the original coordinate system with axes x_1 and x_2. If the data is projected on each of the two axes, the clusters corresponding to the two classes overlap and the classes cannot be separated using any single dimension.

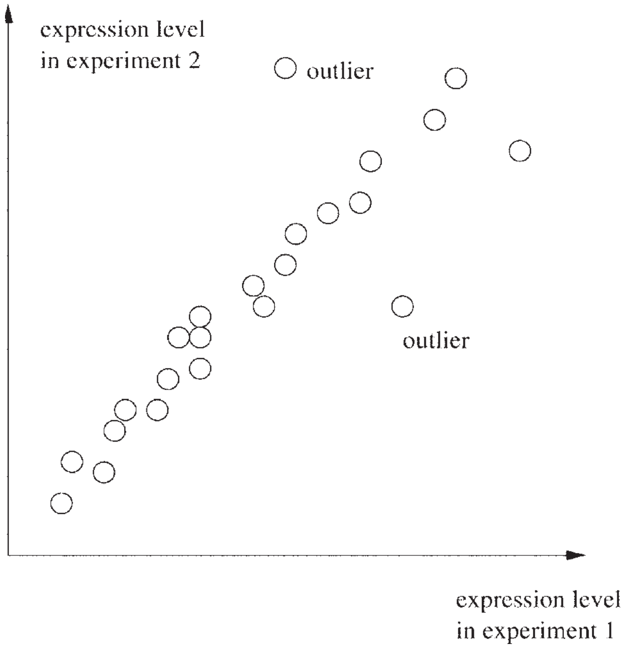


Fig. 3. Expression levels in two experiments visualized as a scatterplot. Points above the first diagonal represent genes with expression levels higher in the second experiment whereas points below the diagonal represent genes with expression levels higher in the first experiment. It is very easy to identify genes that behave differently from one experiment to another.

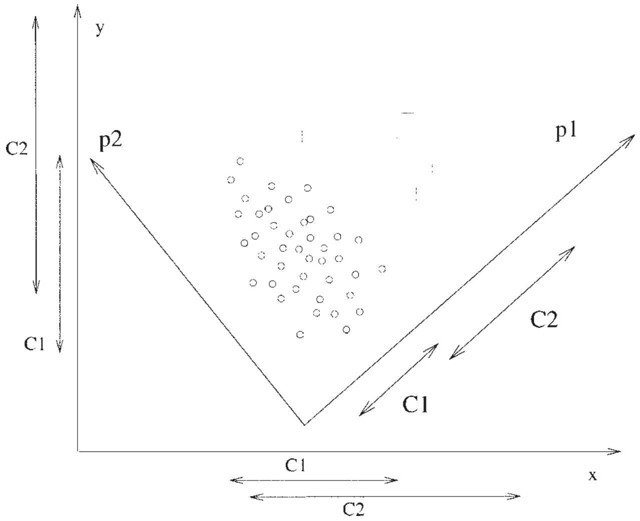


Fig. 4. Principal Component Analysis (PCA). If one of the two original axes *x* or *y* is eliminated, the classes cannot be separated. The co-ordinate system found by PCA (*p*1, *p*2) allows the elimination of *p*2 while preserving the ability to separate the given classes. It is said that PCA performs a dimensionality reduction.

A dimensionality reduction is not possible since both x_1 and x_2 are required to separate the classes.

However, the PCA approach can analyze the data to extract a new coordinate system with axes p_1 and p_2. If the two clusters are projected on the axes of the new coordinate systems one can notice that the situation is different. The projections of the classes on p_2 overlap completely. However, the projections of the two classes on p_1 yield two clusters that can be easily separated. In these conditions, one can discard the second coordinate p_2. In effect, we have achieved a dimensionality reduction from a space with two dimensions to a space with just one dimension while retaining the ability to separate the two classes.

PCA has been shown to be effective in many practical problems including gene expression data and is currently available in a number of software tools. The main limitations of PCA are related to the fact that it only takes into consideration first-order statistical characteristics of the data. The axes of the new coordinate system are the eigenvectors[8] of the covariance matrix of the patterns. In effect, the transformation provided by the change from the original coordinate system to PCA, is only a rotation followed by a scaling proportional to the eigenvalues of the covariance matrix. PCA only considers the variance and not the classes of data so its results are not always as useful as in the earlier example.

Cluster Analysis

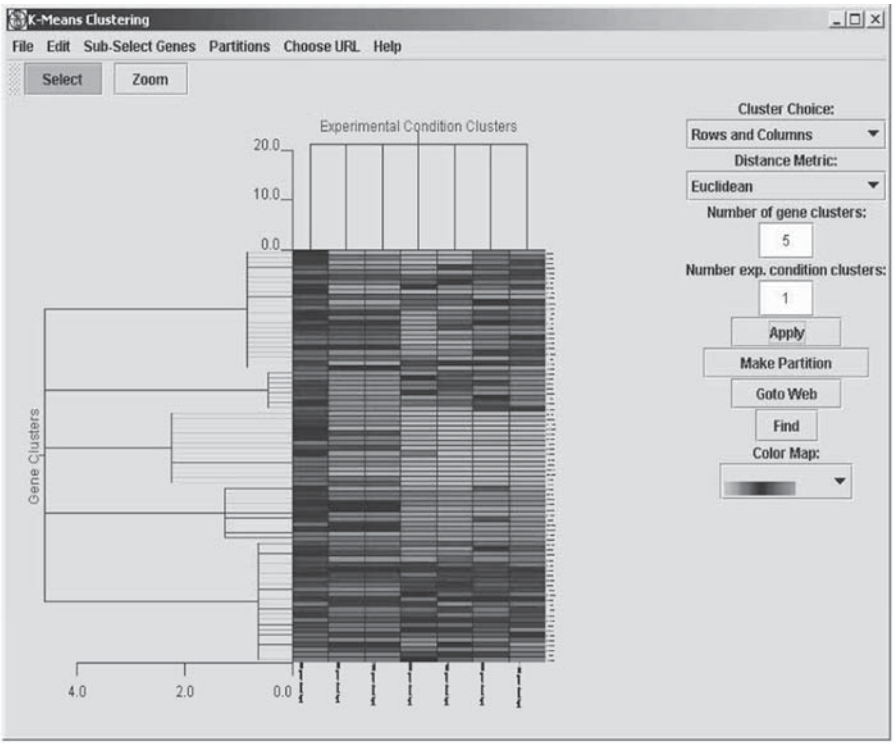
Two frequently posed problems related to microarrays are finding groups of genes with similar expression profiles across a number of experiments and finding groups of individuals with similar expression profiles within a population. The technique most commonly used for such purposes is cluster analysis.

Cluster analysis techniques are essentially dependent on the definition of a metric in the multidimensional space of interest. In practice this means establishing a way to quantitatively calculate how similar two given expression profiles are. Distances such as Euclidean, correlation, squared Euclidean or Manhattan are frequently used. Once such a distance is chosen, one can use it to derive a set of clusters including the given data points. Traditionally, microarray data have been analyzed using hierarchical clustering. Such techniques yield trees of clusters also known as dendrograms. Different genes are grouped together in clusters using the distance chosen. Different clusters are also linked together to form the dendrogram based on a cluster distance such as the average distance between all pairs of objects in the clusters. A combined dendrogram with gene clustering plotted horizontally and experiment clustering plotted vertically is presented in Fig. 5.

K-Means Clustering

The k-means algorithm is often used because it is very simple and very fast. Like any clustering, it can be used to group genes or experiments or any set of homogeneous entities described by a vector of numbers. We shall denote such entities as patterns. Similar patterns grouped together by the algorithm are clusters. A set of clusters including all genes or experiments considered form a clustering.

[8] An eigenvector of a matrix A is defined as a vector x such as $Ax = \lambda x$ where λ is a scalar. Eigenvectors represent a set of directions of the space that are only scaled (i.e., not rotated) by the linear transformation represented by the matrix.



Cluster Choice: Rows and Columns

Distance Metric: Euclidean

Number of gene clusters: 5

Number exp. condition clusters: 1

Fig. 5. K-means clustering in GeneSight. The user can choose the number of clusters as well as the distance used. The colors of the rectangular cells represents the expression values while the colors of the tree branches are associated to specific gene functions. The program also allows inferring gene function for previously unknown genes by functional enrichment analysis.

In k-means clustering the user chooses the value of *k*, which is the number of clusters to be formed. The program starts by randomly choosing *k* points as the centers of the clusters (*see* Fig. 6A). For each pattern, the program will calculate the distance from it to all cluster centers. The pattern will be associated with the closest cluster center. A first approximate clustering is obtained after allocating each pattern to a cluster. However, because the cluster centers were chosen randomly, it is not said that this is the correct clustering. The second step starts by considering all patterns associated to one cluster center and calculating a new position for this cluster center (*see* Fig. 6B). The coordinates of this new center are usually obtained by calculating the mean of the coordinates of the points belonging to that cluster. Since the centers have moved, the pattern membership needs to be updated by recalculating the distance from each pattern to the new cluster centers (*see* Fig. 6C, two patterns move from one cluster to the other). The algorithm continues to update the cluster centers based on the new membership and update the membership of each pattern until the cluster centers are such that no pattern moves from one cluster to another. Since no pattern has changed membership, the centers will remain the same and the algorithm can terminate (*see* Fig. 6D).

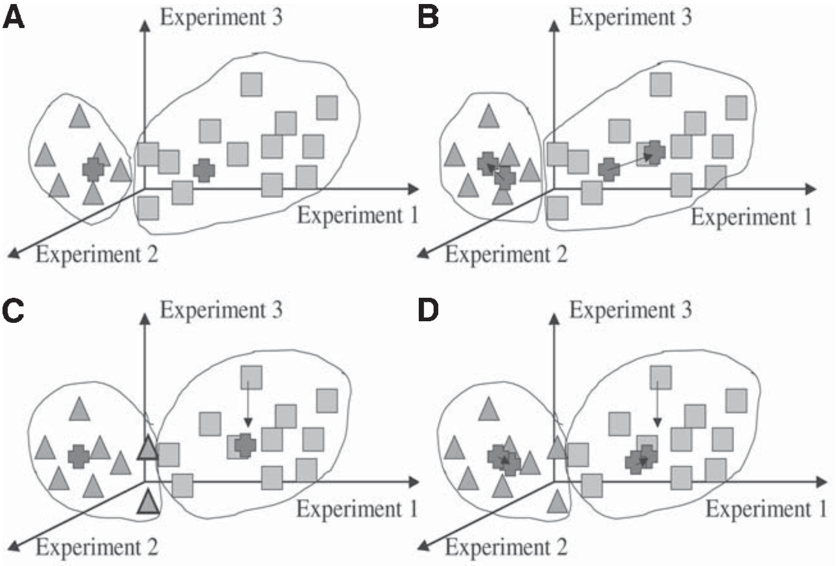


Fig. 6. The k-means algorithm with $k = 2$. **(A)** Two cluster centers are chosen randomly and patterns are assigned to each cluster based on their distance to the cluster center. **(B)** New centers are calculated based on the patterns belonging to each cluster. **(C)** Patterns are assigned to new cluster centers based on distance. Two patterns move from the right cluster to the left cluster. **(D)** New cluster centers are calculated based on the patterns in each cluster. The algorithm will continue trying to reassign patterns. No pattern will be moved between clusters. In consequence, the centers will not move in the next update and the algorithm will stop.

The k-means algorithm has several important properties. First of all, the results of the algorithm, i.e., the clustering or the membership of various patterns to various clusters, *can change* between successive runs of the algorithm (*see* Fig. 7). Furthermore, if some clusters are initialized with centers far from all patterns, no patterns will fall into their sphere of attraction and they will produce empty clusters. In order to alleviate these problems, care should be taken in the initialization phase. A common practice initializes centers with k points chosen randomly from the existing patterns. This ensures that: 1) the starting cluster centers are in the general area populated by the given data, and 2) each cluster will have at least one pattern. This is because if a pattern is initialized as a center of a cluster, it will probably remain in that cluster.

A natural question arises regarding the meaning of the k-means clustering results: if k-means can produce different clusters every time, what confidence can one have in the results of the clustering? This question can be refined into a number of questions that will be briefly considered in the following.

One such question is how good is a particular cluster? One way to assess the goodness of fit of a given clustering is to compare the size of the clusters with the intercluster distance. If the intercluster distance is much larger than the size of the clusters, the clustering is deemed to be more trustworthy (*see* Fig. 8). Therefore, for each cluster, the ratio between the distance D to the nearest cluster center and its diameter d can be used as an indication of the cluster quality. In the dendrogram plotted by

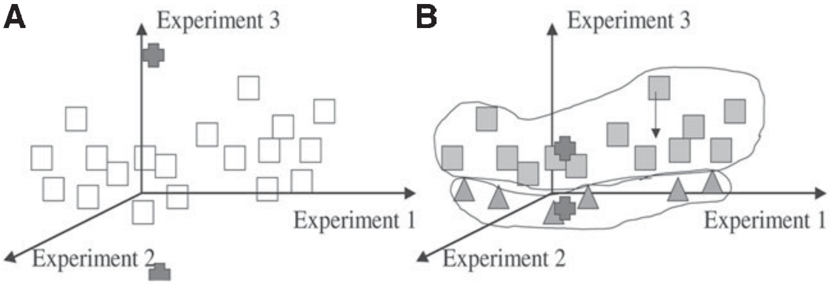


Fig. 7. K-means can produce different results in different runs depending on the initialization of the cluster centers. For instance, **(A)** if the clusters are initialized, **(B)** the final result will be very different from the one obtained previously (in Fig. 6.).

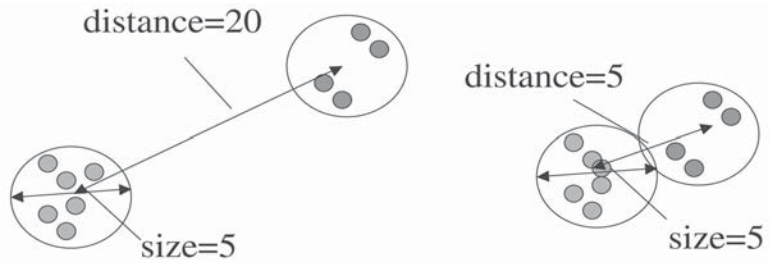


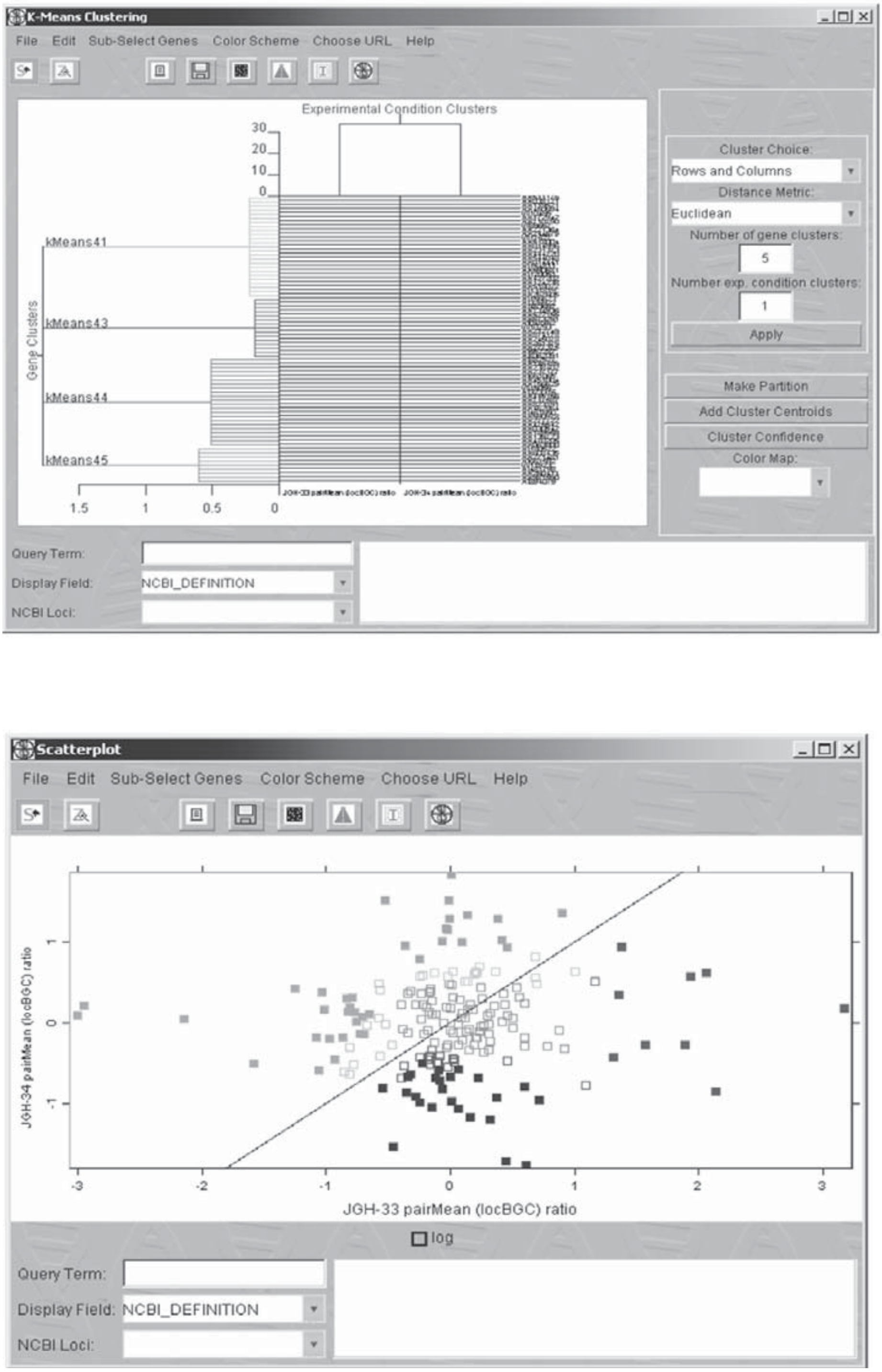
Fig. 8. Cluster quality assessment: clusters for which the nearest adjacent cluster is situated at a distance much larger than their own size are more trustworthy.

GeneSight (*see* Fig. 5), the length of the branches of the tree are proportional to the ratio $D = d$. Thus, *taller* clusters are better than *short* clusters[9].

Another interesting question is how confident can one be that a gene that fell into a cluster will fall into the same cluster if the clustering is repeated? This question can be addressed by repeating the clustering several times and following the particular gene of interest. In GeneSight, this can be achieved by generating a partition based on a given clustering. Such a partition will have all genes in a cluster coded with the same color (*see* Fig. 9). Repeating the clustering several times will reveal whether the colors remain grouped together. Those genes that are clustered together repeatedly are more likely to have a genuine connection. This idea can be taken further to its natural extension that is bootstrapping.

Bootstrapping is a general technique that allows the computation of some goodness of fit measure based on many repeats of the same experiment on slightly different datasets all constructed from the available data. The bootstrap is a very powerful method but is computationally intensive often requiring hours or days of computing time on state-of-the-art machines.

[9] Note that this is specific to GeneSight. Other data analysis programs may not calculate such quality information and therefore the length of the branches in many publications may have no particular significance.



Cluster Choice: Rows and Columns

Distance Metric: Euclidean

Number of gene clusters: 5

Number exp. condition clusters: 1

Query Term:

Display Field: NCBI_DEFINITION

NCBI Loci:

Query Term:

Display Field: NCBI_DEFINITION

NCBI Loci:

Fig. 9. An example of a partition defined by a clustering. Genes falling in one cluster are coded with the same color. Repeating the clustering and choosing those colors that tend to stay together increases the confidence that those genes truly share important features.

The choice of the number of clusters is another issue that needs careful consideration. If it is known in advance that the patterns to be clustered belong to several different classes, e.g., benign and malignant, one should then cluster using the known number of classes. Thus, if there are features that clearly distinguish between the classes, the algorithm might use them to construct meaningful clusters. It is not necessary to know which pattern belongs to each class but only that there are two different classes. If the number of existing classes is not known and the analysis has an exploratory character, one could repeat the clustering for several values of k and compare the results, i.e., track the genes that tend to fall in the same cluster all the time. This approach is heuristic in nature and its utility will vary widely depending on the particular problem studied.

The complexity of the k-means algorithm must also be considered. It can be shown that the k-means algorithm is linear in the number of patterns, e.g., genes, N. This means that the number of computations that need to be performed can be written as $p \cdot N$, where p is a value that does not depend on N but does depend on the number k of clusters chosen by the user as well as the number of iterations. However, the number of clusters is very small, e.g., 10–20 clusters, in comparison with the number of patterns, e.g., 30,000 genes. One can conclude that k-means has a very low computational complexity which translates directly into a high speed.

Hierarchical Clustering

Hierarchical clustering aims at the more ambitious task of providing *the* clustering that characterizes a set of patterns in the context of a given distance metric. The result of the clustering will be a complete tree with individual patterns, genes or experiments as leaves and the root as the convergence point of all branches. The tree can be constructed in a bottom-up fashion, starting from the individual patterns and working upwards towards the root[10] or following a top-down approach, starting at the root and working downwards towards the leaves. The bottom-up approach is sometimes called agglomeration because it works by putting smaller clusters together to form bigger clusters. Analogously, the top-down approach is sometime called division because it works by splitting large clusters into smaller ones.

Unlike k-means, a hierarchical clustering algorithm applied on a given data set and using a chosen distance will always produce the same tree. However, different hierarchical clustering algorithms, e.g., bottom-up and top-down, may produce different trees. In general, algorithms working by division require less computation and are therefore faster. However, obtaining the results quicker may not necessarily be a reason for joy because a hierarchical clustering algorithm working by division, or top-down, may produce results worse than an algorithm working by agglomeration. This can happen because in dividing the clusters the most important splits, affecting many patterns, are performed at the beginning before accumulating enough information and two patterns inadvertently placed in different clusters by an early splitting decision will never be put together again.

The bottom-up method works as follows. Start with n clusters, each consisting of a single point (gene or experiment). Calculate a table containing the distances from each cluster to every other cluster (for n points this will be of the order of n^2 computations). Then repeatedly merge the two most similar clusters into a single superclus-

[10] Unlike the real trees, classification trees are usually drawn with the root on top and the branches developing underneath.

ter until the entire tree is constructed. A cluster distance is required to assess the similarity between clusters.

The distance between clusters can be taken to be the distance between the closest neighbors (known as single linkage clustering) furthest neighbors (complete linkage), the distance between the centers of the clusters (centroid linkage), or the average distance of all patterns in each cluster (average linkage). Clearly, the total complexity of the algorithm and therefore its speed is very much dependent on the linkage choice. Single or complete linkages require only choosing one of the distances already calculated while more elaborated linkages such as centroid, require more computations. Such further computations are needed *every time two clusters are joined*, which greatly increases the total complexity of the clustering. However, cheap is not always better. Simple and fast methods such as single linkage tend to produce long stringy clusters, when e.g., using a Euclidean distance, while more complex methods such as centroid linkage or neighbor joining tend to produce clusters that more accurately reflect the structure present in the data but are extremely slow. The complexity of a bottom-up implementation can vary between n^2 and n^3 depending on the linkage chosen. In the context of gene expression, one should try to prune the set of genes of interest as much as possible before attempting to apply bottom-up clustering with a more complex linkage.

The top-down approach starts by considering the whole set of patterns to be clustered. Subsequently, the algorithm uses any of a large number of nonhierarchical clustering algorithms to divide the set into two clusters. A particular choice of such a nonhierarchical algorithm is k-means with $k = 2$. Subsequently, the process is recursively repeated on each of the smaller clusters as they are obtained. The process stops when all small clusters contain a single pattern.

The top-down clustering tends to be faster but the clusters produced tend to less accurately reflect the structure presented in the data. Theoretically, the results of a hierarchical clustering should only depend on the data and the metric chosen. However, a top-down approach will essentially rely on the qualities of the partitioning algorithm. For example, if k-means is chosen to divide clusters into subclusters, the overall result may be different if the algorithm is run twice with the same data. This can be due to the random initialization of the cluster centers in the k-means division. The complexity of the top-down approach can require between *nlogn* and n^2 computations and is therefore intrinsically faster than the bottom-up approach especially when a complex linkage is involved.

Finally, another approach to building a hierarchical cluster uses an incremental method. This approach can be even faster than the top-down approach. Such methods build the dendrogram by adding one point at a time, with minimal changes to the existing hierarchy. In order to add a new gene, the gene under consideration is compared with each cluster in the tree, starting with the root and following always the most similar branch according to the distance used. When finding a cluster containing a single gene, the algorithm adds a branch containing the gene under consideration. This approach can be lightning fast compared to the others. However, the weakness is that the results can depend not only on the distance metric (as any clustering) or the distance metric and some random initialization (as the top-down approach) but also on the *order* in which the points are considered.

A few conclusions can be drawn from this discussion of various hierarchical clustering methods. First, various implementations using the same data and the same metric can still produce different dendrograms if they use different approaches. Second, merely obtaining a clustering is not, nor is the dendrogram itself the answer to the researcher's desires. A dendrogram connecting various genes in a graphically pleasant way can be obtained relatively quickly even if thousands of genes are involved. The real problem is to obtain a clustering that reflects the structure of the data. A clustering that reflects well the properties of the data may require more computations. Therefore, various implementations of hierarchical clustering should not be judged simply by their speed. Many times, slower algorithms may simply be trying to do a better job of extracting the data features.

ANOVA

In order to account for the multiple sources of variation in a microarray experiment, Kerr and Churchill used an ANOVA approach and proposed the following model:

$$\log(y_{ijkg}) = \mu + A_i + D_j + V_k + G_g + (AG)_{ig} + (VG)_{kg} + \varepsilon_{ijkg} \qquad \text{[Eq. 1]}$$

In this model, μ is the overall mean signal of the array, A_i is the effect of the i^{th} array, D_j represents the effect of the j^{th} dye, V_k represents the effect of the k^{th} variety (a condition such as healthy or malignant), G_g represents the variation of the g^{th} gene, $(AG)_{ig}$ is the effect of a particular spot on a given array, $(VG)_{kg}$ represents the interaction between the k^{th} variety and the g^{th} gene and ε_{ijkg} represents the error term for array i, dye j, variety k, and gene g. The error is assumed to be independent and of zero mean.

The ANOVA method uses a classical optimization algorithm to find numerical values that fit the given data. ANOVA is usually followed by a residual analysis, in which one studies the residuals, i.e., the differences between the measured values and the values given by the model. The purpose of this analysis is to show that the model is appropriate. If the distribution of the residuals is consistent with a random distribution, the approach is considered to be successful and the model used suitable for the given phenomenon. If the residuals show any particular shape that is not consistent with a random distribution, the model is considered to be inadequate for the given phenomenon. The rationale behind this analysis is that if the model is accurately representing the phenomenon, the differences between the predicted values and the real values should be only due to the noise that is random. If the residuals are not random, the data contained some feature that is not properly represented in the model and thus is visible in the residuals.

The ANOVA has several important advantages. First, it provides an explicit quantitative term for each factor considered in the noise. Thus, differences between different arrays as well as dye effects can be precisely assessed and normalization becomes unnecessary. Second, it is easy to find differentially regulated genes in a rigorous manner because there is an explicit term that relates to the expression level of the genes.

Among the disadvantages, ANOVA is very sensitive to the experimental design. If the experiment is not carried out in a manner that is consistent with the model, various factors will be confounded. In spite of this, the ANOVA is one of the best methods currently in use for analysis of microarray data.

Supervised Learning

Cluster analysis is currently by far the most frequently used multivariate technique to analyze gene sequence expression data. Clustering is appropriate when there is no *a priori* knowledge about the data. In such circumstances, the only possible approach is to study the similarity between different samples or experiments. In a machine learning framework, this process is known as unsupervised learning since there is no known desired answer for any gene or experiment.

However, we are now starting to understand the way genes function. The information has started to be organized and there are public databases storing functional information and pathways describing gene interactions. Such information may be used in the data analysis process. For example, the clustering step can be followed by a step in which one attaches functional labels to various clusters. This can be based on the members of the clusters whose function is known. Subsequently, one can construct a number of hypotheses that attach functionality to genes whose function is not currently known. Such hypotheses can then be tested. This mechanism is an effective way to construct a set of reasonable hypotheses from a large number of expression values and may be the key to establishing structure-function relationships. Furthermore, this approach permits the construction of metabolic pathways and the discovery of previously unknown functional relationships.

The disadvantage of clustering techniques is related to the fact that there are no guarantees that the clusters obtained make sense from a biological point of view. In other words, there is no guarantee that there is a clear biological function attached to each cluster.

Recently, a number of laboratories addressed the issue of using gene expression to distinguish between classes of conditions such as leukemia, colon cancer, malignant melanoma, and B-cell lymphoma. In a machine learning framework, this modus operandi is known as supervised learning and is very suitable for identifying the function of new genes.

Reviews of various supervised and unsupervised techniques are presented as well as other results using supervised learning (*see* Suggested Readings). Other classical machine learning techniques such as nearest neighbor, boosting, Bayesian classifiers, support vector machines, singular value decomposition have shown to be successful in microarray data analysis.

Other Existing Research and Tools

Several recent review articles discuss computational methods and challenges in this field. Various data analysis methods have been used to extract functional information. One favorite target for such studies is the yeast genome.

Some software tools have already been developed for the visualization, manipulation and analysis of gene expression data. GenExplore (Applied Maths) can do cluster analysis, PCA, discriminant analysis, and SOFMs. ImaGene and GeneSight (Biodiscovery) offer everything from image processing to data analysis including clustering and PCA. XDotsReader (Cose, France) offers calibration, normalization, and statistic analysis for DNA grids. WorkBench, Phylosopher, Impressionist, and Expressionist (GeneData AG) offer gene expression, sequence comparison, protein expression analysis, and data management tools. GeneCluster (MIT's Whitehead Genome Center); ArrayVision (Imaging Research Inc.); arraySCOUT (LION Bioscience AG); Stingray (Affymetrix); ArrayAnalyzerDB (MolecularWare, Inc.); Resolver (Rosetta Inpharmatics); the MicroArray Suite (Scanalytics Inc.); GeneSpring (Silicon Genetics); Array-Pro (Spotfire Inc.); Cluster, Xcluster, and TreeView (Stanford); ArrayViewer (TIGR); Axeldb; and MUMmer and Glimmer (IGR) are few other tools offering various visualization and analysis tools for DNA data analysis.

DNA array databases and on-line tools include: GeneX, Arrayexpress, SAGEmap, μArray, ArrayDB, ExpressDB, SMD, KEGG, and MGED (*see* Suggested Readings).

This wealth of software and resources clearly shows the amount of effort currently directed towards developing new tools for microarray data analysis. Microarrays have already proven invaluable tools able to shed new light on subtle and complex phenomena that happen at the genetic level. However, in order to realize its full potential, the laboratory work using microarrays needs to be complemented by a careful and detailed analysis of the results. The early years of this technology were characterized by a wealth of spectacular results obtained with very simple data-analysis tools. Those were some of the low-hanging fruits of the field. A careful analysis performed with suitable tools may reveal that those low-hanging fruits were only a very small fraction of the potential crop. Furthermore, it is likely that the most spectacular results, deeply affecting the way we currently understand the genetic mechanism, are further up in the knowledge tree and their gathering will require a close symbiosis between biologists, computer scientists and statisticians.

Suggested Readings

Microarrays: Tools for Gene Expression Analysis

Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., et al. (2000) Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, Nature 6769(403), 503–511.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999) Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by nucleotide arrays, Proc. Natl. Acad. Sci. USA 96, 6745–6750.

Bittner, M., Meltzer, P., Chen, Y., Jiang, Y., Seftor, E., Hendrix, M., et al. (2000) Molecular classification of cutaneous malignant melanoma by gene expression profiling, Nature 406(6795), 536–540.

Der, S., Williams, B., and Silverman R. (1998) Identification of genes differentially regulated by interferon alpha, beta, or gamma using oligonucleotide arrays, Proc. Natl. Acad. Sci. USA 26(95), 15,623–15,628.

Efron, B. and Tibshirani, R. J. (1993) An introduction to Boostrap, Chapman and Hall, London, UK.

Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA 95, 14,863–14,868.

Fan, J.-B., Chen, X., Halushka, M., Berno, A., Huang, X., Ryder, et al. (2000) Parallel genotyping of human SNPs using generic high-density oligonucleotide tag arrays, Genome Res. 10, 853–860.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286(5439), 531–537.

Khan, J., Saal, L. H., Bittner, M. L., Chen, Y., Trent, J. M., and Meltzer, P. S. (1999) Expression profiling in cancer using cDNA microarrays, Electrophoresis 20(2), 223–229.

Lockhart, D. J., Dong, H., Byrne, M., Folletie, M., Gallo, M. V., Chee, M. S., et al. (1996) DNA expression monitoring by hybridization of high density oligonucleotide arrays, Nat. Biotech. 14, 1675–1680.

Perou, C., Sorlie, T., Eisen, M., van de Rijn, M., Jeffrey, S., Rees, C., et al. (2000) Molecular portraits of human breast tumors, Nature 406(6797), 747–752.

Schena, M. (2000) Microarray Biochip Technology, Eaton Publishing, Natik, MA.

Schena, M., Shalon, D., Davis, R., and Brown, P. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray, Science 270, 467–470.

Shalon, D., Smith, S. J., and Brown, P. O. (1996) A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization, Genome Res. 6, 639–645.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, W., Anders, K., Eisen, M. B., et al. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization, Mol. Biol. Cell 9(12), 3273–3297.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, Proc. Natl. Acad. Sci. USA 96, 2907–2912.

Wang, D. G., Fan, J. B., Siao, C., Berno, J. A., et al. (1998) Large-scale identification, mapping, and genotyping of single-nucleotide polymorphisms in the human genome, Science 280(5366), 1077–1082.

Zhu, H., Cong, J., Mamtora, G., Gingeras, T., and Shenk, T. (1998) Cellular gene expression altered by human cytomegalovirus: global monitoring with oligonucleotide arrays, Proc. Natl. Acad. Sci. USA 24(95), 14,470–14,475.

Experimental Design

Draghici, S., Kuklin, A., Hoff, B., and Shams, S. (2001) Experimental design, analysis of variance and slide quality assessment in gene expression arrays, Curr. Opin. Drug Disc. Dev. 4(3), 332–337.

Fisher, R. (1951) The Design of Experiments, Oliver and Boyd, London, UK.

Kerr, M. K. and Churchill, G. A. (2001) Analysis of variance for gene expression microarray data, J. Comp. Biol. 7(6):819–837.
(*see* Website: http://www.jax.org/research/churchill/pubs/index.html)

Kerr, M. K. and Churchill, G. A. (2001) Experimental design for gene expression analysis, Biostatistics 2, 183–201.
(*see* Website: http://www.jax.org/research/churchill/pubs/index.html)

Kerr, M. K. and Churchill, G. A. (2001) Statistical design and the analysis of gene expression, Genet. Res. 77, 123–128.
(*see* Website: http://www.jax.org/research/churchill/pubs/index.html)

Schuchhardt, J., Beule, D., Wolski, E., and Eickhoff, H. (2000) Normalization strategies for cDNA microarrays, Nucleic Acids Res. 28(10), e47i–e47v.

West, M., Nevins, J., Marks, J., Spang, R., Blanchette, C., and Zuzan, H. (2000) Bayesian regression analysis in the “large p, small n” paradigm with application in DNA microarray studies, Technical report, Duke University Medical Center, Durham, NC.

Data Normalization

Arabidopsis (2001) Normalization method comparison, Technical report, Stanford University, Stanford, CA.
(*see* Website: http://www.afgc.stanford.edu/ finkel/talk.htm)

Bouton, C., Henry, G. W., and Pevsner, J. (2001) Database referencing of array genes online—DRAGON. Technical report, Kennedy Krieger Institute, Baltimore, MD. (*see* Website: http://www.pevsnerlab.kennedykrieger.org/dragon.htm)

Chen, Y., Dougherty, E. R., and Bittner, M. L. (1997) Ratio-based decisions and the quantitative analysis of cDNA microarray images, J. Biomed. Optics 2(4), 364–374.

Cheng, L. and Wong, W. (2001) Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection, Proc. Natl. Acad. Sci. USA 98, 31–36.

Dudoit, S., Yang, Y. H., Callow, M., and Speed, T. (2000) Statistical models for identifying differentially expressed genes in replicated cDNA microarray experiments, Technical Report 578, University of California, Berkeley, CA.

Hedge, P., Qi, R., Abernathy, K., Gay, C., Dharap, S., Gaspard, R., et al. (2000) A concise guide to cDNA microarray analysis, Biotechniques 29(3), 548–562.

Houts, T. M. (2000) Improved 2-color Exponential normalization for microarray analyses employing cyanine dyes, in: Proceedings of CAMDA, Critical Assessment of Techniques for Microarray Data Mining, (Lin, S., ed.), December 18–19, Duke University Medical Center, Durham, NC.

Kepler, T., Crosby L., and Morgan, K. (2001) Normalization and analysis of DNA microarray data by self-consistency and local regression, Nucleic Acids Res., submitted.

Long, A., Mangalam, H., Chan, B., Tolleri, L., Hatfield, G. W., and Baldi, P. (2001) Improved statistical inference from DNA microarray data using analysis of variance and a Bayesian statistical framework, J. Biol. Chem. 276(23), 19,937–19,944.

Schadt, E. E., Cheng, L., Su, C., and Wong, W. H. (2000) Analyzing high-density oligonucleotide gene expression array data, J. Cell. Biochem. 80, 192–202.

Speed, T. P. (2000) Hints and prejudices—always log spot intensities and ratios, Technical report, University of California, Berkeley, CA. (*see* Website: http://www.stat.berkeley.edu/users/terry/zarray/Html/log.html)

Yang, Y., Buckley, M. J., Dudoit, S., and Speed, T. P. (2000) Comparison of methods for image analysis on cDNA, Technical report, University of California, Berkeley, CA. (*see* Website: http://www.stat.berkeley.edu/users/terry/zarray/Html/log.html)

Yang, Y., Dudoit, S., Luu, P., and Speed, T. P. (2000) Normalization for cDNA microarray data, Technical report, University of California, Berkeley, CA. (*see* Website: http://www.stat.berkeley.edu/users/terry/zarray/Html/normspie.html)

Yue, H., Eastman, P., Wang, B., Minor, J., Doctolero, M., Nuttall, R. L., et al. (2001) An evaluation of the performance of cDNA microarrays for detecting changes in global mRNA expression, Nucleic Acids Res. 29(8), e41.

(2001) Imagene User's manual, Technical report, BioDiscovery Inc., Marina del Rey, CA. (*see* Website: http://www.BioDiscovery.com)

Selection of Differentially Regulated Genes and Significance Analysis

Aharoni, A., Keizer, L. C. P., Bouwneester, H. J., Sun, Z., Alvarez-Huerta, M., Verhoeven, H. A., et al. (1975) Identification of the SAAT gene involved in strawberry flavor biogenesis by use of DNA microarrays, Plant Cell 12, 647–661.

Audic, S. and Claverie, J.-M. (1998) Vizualizing the competitive recognition of TATA-boxes in vertebrate promoters, Trends Genet. 14, 10–11.

Baldi, P. and Long, A. D. (2001) A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes, Bioinformatics 17(6), 509–519.

Bonferroni, C. E. (1935) Il calcolo delle assicurazioni su gruppi di teste, chapter Studi in Onore del Professore Salvatore Ortu Carboni, Rome, Italy, pp. 13–60.

Bonferroni, C. E. (1936) Teoria statistica delle classi e calcolo delle probabilità, Pubblicazioni del Istituto Superiore di Scienze Economiche e Commerciali di Firenze 8, 3–62.

Brazma, A. and Vilo, J. (2000) Gene expression data analysis, Feder. Euro. Biochem. Soc. Lett. 480(23893), 17–24.

Brown, C. C. and Fears, T. R. (1981) Exact significance levels for multiple binomial testing with application to carcinogenicity screens, Biometrics, 37, 763–774.

Claverie, J.-M. (1999) Computational methods for the identification of differential and coordinated gene expression, Human Mol. Genet. 8(10), 1821–1832.

DeRisi, J. L., Iyer, V. R., and Brown, P. O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale, Science 278, 680–686.

DeRisi, J. L., Penland, L., Brown, P. O., Bittner, M. L., Meltzer, P., Ray, M., et al. (1996) User of a cDNA microarray to analyse gene expression patterns in human cancer, Nature Genet. 14(4), 457–460.

D'haeseller, P. (2000) Genetic Network Inference: From Co-Expression Clustering to Reverse Engineering, PhD Thesis, University of New Mexico, Albuquerque, NM.

D'haeseller, P., Liang, S., and Somogyi, R. (2000) Genetic network inference: From co-expression clustering to reverse engineering, Bioinformatics 8(16), 707–726.

Draghici, S., Cumberland, L., and Kovari, L. C. (2000) Correlation of HIV protease structure with Indinavir resistance, a data mining and neural network approach, Proc. SPIE 4057.

Felsenstein, J. (1985) Confidence limits on phylogenies: an approach using the bootstrap, Evolution 39, 783–791.

Galitski, T., Saldanha, A. J., Styles, C. A., Lander, E. S., and Fink, G. R. (1999) Ploidy regulation of gene expression, Science 285, 251–254.

Hastie, T., Tibshirani, R., Eisen, M. B., Alizadeh, A., Levy, R., Staudt, L., et al. (2000) 'Gene shaving' as a method for indentifying distinct sets of genes with similar expression patterns. Genome Biol. 1(2), 1–21.

Heyse, J. and Rom, D. (1988) Adjusting for multiplicity of statistical tests in the analysis of carcinogenicity studies, Biometr. J. 30, 883–896.

Hill, A. A., Hunter, C. P., Tsung, B. T., Tucker-Kellogg, G., and Brown, E. L. (2000) Genomic analysis of gene expression in *C. elegans*, Science 290, 809–812.

Hochberg, Y. and Tamhane, A. C. (1987) Multiple comparison procedures, John Wiley and Sons, New York, NY.

Holland, B. and Copenhaver, M. D. (1987) An improved sequentially rejective Bonferroni test procedure, Biometrica 43, 417–423.

Holm, S. (1979) A simple sequentially rejective multiple test procedure, Scand. J. Statistics 6, 65–70.

Kerr, M. K. and Churchill, G. A. (2001) Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments, Proc. Natl. Acad. Sci. USA 98(16), 8961–8965.
(*see* Website: http://www.jax.org/research/churchill/pubs/index.html)

Lee, M.-L. T., Kuo, F. C., Whitmore, G. A., and Sklar, J. (2000) Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations, Proc. Natl. Acad. Sci. USA 97(18), 9834–9839.

Manduchi, E., Grant, G. R., McKenzie, S. E., Overton, G. C., Surrey, S., and Stoeckert, C. J. (2000) Generation of patterns from gene expression data by assigning confidence to differentially expressed genes, Bioinformatics 16(8), 685–698.

Newton, M., Kendziorski, C., Richmond, C., Blattner, F. R., and Tsui, K. (1999) On differential variability of expression ratios: improving statistical inference about gene expresison changes from microarray data, Technical report, University of Wisconsin, Madison, WI.
(*see* Website: http://www.biostat.wisc.edu/geda/eba.html)

Richmond, C. S., Glasner, J. D., Mau, R., Jin, H., and Blattner, F. R. (1999) Genome-wide expression profiling in *Escherichia coli* K-12, Nucleic Acids Res. 27(19), 3821–3835.

Roberts, C. J., Nelson, B., Marton, M. J., Stoughton, R., Meyer, M. R., Bennett, H. A., et al. (2000) Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles, Science 287, 873–880.

Sapir, M. and Churchill, G. A. (2000) Estimating the posterior probability of differential gene expression from microarray data, Technical Report, Jackson Labs, Bar Harbor, ME.
(*see* Website: http://www.jax.org/research/churchill/pubs/)

Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P., and Davis, R. (1996) Parallel human genome analysis: microarray-based expression monitoring of 1000 genes, Proc. Natl. Acad. Sci. USA 93, 10,614–10,519.

Shaffer, J. P. (1986) Modified sequentially rejective multiple test procedures, J. Am. Statistical Assoc. 81, 826–831.

Shaffer, J. P. (1995) Multiple hypothesis testing, Ann. Rev. Psychol. 46, 561–584.

Sudarsanam, P., Iyer, V. R., Brown, P. O., and Winston, F. (2000) Whole-genome expression analysis of snf/swi mutants of *Saccharomyces cerevisiae*, Proc. Natl. Acad. Sci. USA 97(7), 3364–3369.

Tao, H., Bausch, C., Richmond, C., Blattner, F. R., and Conway, T. (1999) Functional genomics: expression analysis of *Escherichia coli* growing on minimal and rich media, J. Bacteriol. 181(20), 6425–6440.

ter Linde, J. J., Liang, H., Davis, R. W., Steensma, H. Y., Dijken, J. P., and Pronk, J. T. (1999) Genome-wide transcriptional analysis of aerobic and anaerobic chemostat cultures of *Saccharomyces cerevisiae*, J. Bacteriol. 181(24), 7409–7413.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001) Significance analysis of microarrays applied to the ionizing radiation response, Proc. Natl. Acad. Sci. USA 98(9), 5116–5121.

Wellmann, A., Thieblemont, C., Pittaluga, S., Sakai, A., Jaffe, E. S., Seibert, P., and Raffeld, M. (2000) Detection of differentially expressed genes in lymphomas using cDNA arrays: identification of *clusterin* as a new diagnostic marker for anaplastic large-cell lymphomas, Blood 96(2), 398–404.

Westfall, P. H. and Young, S. S. (1993) Resampling-Based Multiple Testing: Examples and Methods for *p*-Value Adjustment, John Wiley and Sons, New York, NY.

White, K. P., Rifkin, S. A., Hurban, P., and Hogness, D. S. (1999) Microarray analysis of Drosophila development during metamorphosis, Science 286, 2179–2184.

Data Analysis Techniques

Aach, J., Rindone, W., and Church, G. M. (2000) Systematic management and analysis of yeast gene expression data, Genome Res. 10, 431–445.
(*see* Website: http://www.arep.med.harvard.edu/ExpressDB)

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999) Clustering gene expression patterns, J. Comp. Biol. 6(3/4), 281–297.

Brazma, A. (1998) Mining the yeast genome expression and sequence data, BioInformer (4).
(*see* Website: http://www.bioinformer.ebi.ac.uk/newsletter/archives/4/lead article.html)

Ewing, R. M., Kahla, A. B., Poirot, O., Lopez, F., Audic, S., and Claverie, J.-M. (1999) Large-scale statistical analyses of rice ESTs reveal correlated patterns of gene expression, Genome Res. 9, 950–959.

Getz, G., Levine, E., and Domany, E. (2000) Coupled two-way clustering analysis of gene microarray data, Proc. Natl. Acad. Sci. USA 97(22), 12,079–12,084.

Herwig, R., Poustka, A., Muller, C., Bull, C., Lehrach, H., and O'Brien, J. (1999) Large-scale clustering of cDNA-fingerprinting data, Genome Res. 9(11), 1093–1105.

Heyer, L. J., Kruglyak, S., and Yooseph, S. (1999) Exploring expression data: identification and analysis of coexpressed genes, Genome Res. 9, 1106–1115.

Hilsenbeck, S., Friedrichs, W., Schiff, R., O'Connell, P., Hansen, R., Osborne, C., and Fuqua, S. W. (1999) Statistical analysis of array expression data as applied to the problem of Tamoxifen resistance, J. Natl. Cancer Inst. 91(5), 453–459.

Pietu, G., Mariage-Samson, R., Fayein, N.-A., Matingou, C., Eveno, E., Houlgatte, R., et al. (1999) The Genexpress IMAGE knowledge base of the human brain transcriptome: a prototype integrated resource for functional and computational genomics, Genome Res. 9, 195–209.

Raychaudhuri, S., Stuart, J. M., and Altman, R. (2000) Principal components analysis to summarize microarray experiments: application to sporulation time series, Proc. Pacific Symp. Biocomp. 5, 452–463.

Sidák, Z. (1967) Rectangular confidence regions for the means of multivariate normal distributions, J. Am. Statistical Assoc. 62, 626–633.

Souvaine, D. L. and Steele, J. M. (1987) Efficient time and space algorithms for least median of squares regression, J. Am. Statistical Assoc. 82, 794–801.

Tsoka, S. and Ouzounis, C. A. (2000) Recent developments and future directions in computational genomics, Feder. Euro. Biochem. Soc. Lett. 23897, 1–7.

van Helden, J., Rios, A. F., and Collado-Vides, J. (2000) Discovering regulatory elements in non-coding sequences by analysis of spaced dyads, Nucleic Acids Res. 28(8), 1808–1818.

Wang, M. L., Belmonte, S., Kim, U., Dolan, M., Morris, J. W., and Goodman, H. M. (1999) A cluster of ABA-regulated genes on *Arabidopsis Thaliana* BAC T07M07, Genome Res. 9, 325–333.

Zhang, M. Q. (1999) Large-scaled gene expression data analysis: a new challenge to computational biologists, Genome Res. 9, 681–688.

Zhu, J. and Zhang, M. (2000) Cluster, function and promoter: analysis of yeast expression array, Pac. Symp. Biocomp., 476–487.

Supervised Learning

Alter, O., Brown, P., and Botstein, D. (2000) Singular value decomposition for genome-wide expression data processing and modeling, Proc. Natl. Acad. Sci. USA 97(18), 10,101–10,106.

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, Nature Genet. 25, 25–29.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., and Yakhini, Z. (2000) Tissue classification with gene expression profiles, in: Proceedings of the Fourth Annual Interternational Conference on Computational Molecular Biology (RECOMB 2000), Tokyo, Japan, pp. 54–64.

Brazma, A., Jonassen, I., Eidhammer, I., and Gilbert, D. (1998) Approaches to the automatic discovery of patterns in biosequences, J. Comp. Biol. 5(2), 279–305.

Brown, M. P. S., Grundy, W. B., Lin, D., Christianini, N., Sugnet, C. W., Ares, M., and Haussler, D. (1999) Support vector machine classification of microarray gene expression data, Technical report, University of California, Santa Cruz, CA.

Brown, M. P. S., Grundy, W. B., Lin, D., Cristianini, N., Sugnet, C. W., Furgey, T. S., et al. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Natl. Acad. Sci. USA 97(1), 262–267.

Butte, A. and Kohane, I. (2000) Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements, Pac. Symp. Biocomp., 418–429.

Butte, A., Tamayo, P., Slonim, D., Golub, T., and Kohane, I. (2000) Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks, Proc. Natl. Acad. Sci. USA 97(22), 12,182–12,186.

Celis, J. E., Kruhoffer, M., Gromova, I., Frederiksen, C., Ostergaard, M., Thykjaer, T., et al. (2000) Gene expression profiling: monitoring transcription and translation products using DNA microarrays and proteomics, Fed. Euro. Biochem. Soc. Lett. 23892, 1–15.

Cho, R., Huang, M., Campbell, M., Dong, H., Steinmetz, L., Sapinoso, L., et al. (2001) Transcriptional regulation and function during the human cell cycle, Nat. Genet. 27, 48–54.

Cortes, C. and Vapnik, V. (1995) Support-vector networks, Machine Learn. 20(3), 273–297.

Duda, R. and Hart, P. (1973) Pattern Classification and Scene Analysis, John Wiley and Sons, New York, NY.

Freung, Y. (1997) A decision-theoretic generalization of on-line learning and an application to boosting, J. Comp. System Sci. 55, 119–139.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., and Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16(10), 906–914.

Keller, A., Shummer, M., Hood, L., and Ruzzo, W. (2000) Bayesian classification of DNA array expression data, Technical report UW-CSE-2000-08-01, University of Washington, Seattle, WA.

Stitson, M. O., Weston, J. A. E., Gammerman, A., Vovk, V., and Vapnik, V. (1996) Theory of support vector machines, Technical report CSD-TR-96-17, Royal Holloway University, London, UK.

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., and Church, G. M. (1999) Systematic determination of genetic network architecture, Nat. Genet. 22, 281–285.

(2001) Gene Ontology, Technical report, Gene Ontology Consortium.
(*see* Website: http://www.geneontology.org/)

(2001) Kyoto Encyclopedia of Genes and Genomes, Technical report, Kyoto University, Kyoto, Japan.
(*see* Website: http://www.genome.ad.jp/kegg/)

(2001) National Center for Biotechnology Information (NCBI), Technical report, National Library of Medicine, National Institutes of Health, Bethesda, MD.
(*see* Website: http://www.ncbi.nlm.nih.gov/)

Other Existing Research and Tools

Anderson, J. S. J. and Parker, R. (2000) Computational identification of cis-acting elements affecting post-transcriptional control of gene expression in *Saccharomyces cerevisiae*, Nucleic Acids Res. 28(7), 1604–1617.

Bassett, Jr., D., Eisen, M. B., and Boguski, M. S. (1999) Gene expression informatics—it's all in your mind, Nat. Genet. 21(suppl), 51–55.

Brazma, A., Jonassen, I., Vilo, J., and Ukkonen, E. (1998) Predicting gene regulatory elements in silico on a genomic scale, Genome Res. 8, 1202–1215.

Colello, G., Stewart, J., Zhou, J., Montoya, L., Mangalam, H., Pear, M. R., and Peterson, T. (2001) GeneX, Technical report, National Center for Genome Research, Sante Fe, NM.
(*see* Website: http://www.ncgr.org/genex/team.html)

Hieter, P. and Boguski, M. (1997) Functional genomics: it's all how you read it, Science 278(5338), 601–602.

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., et al. (2000) Functional discovery via a compendium of expression profiles, Cell 102, 109–126.

Jansen, R. and Gerstein, M. (2000) Analysis of the yeast transcriptome with structural and functional categories: characterizing highly expressed proteins, Nucleic Acids Res. 28(6), 1481–1488.

Kal, A. J., van Zonneveld, A. J., Benes, V., van den Berg, M., Koerkamp, M. G., Albermann, K., et al. (1999) Dynamics of gene expression revealed by comparison of serial analysis of gene expression transcript profiles from yeast grown on two different carbon sources, Mol. Biol. Cell 10, 1859–1872.

Lash, A. E., Tolstoshev, C. M., Wagner, L., Shuler, G. D., Strausberg, R. L., Riggins, G. J., and Altschul, S. F. (2000) SAGEmap: a public gene expression resource, Genome Res. 10, 1051–1060.

Nierras, C. R. and Warner, J. R. (1999) Protein Kinase C enables the regulatory circuit that connects membrane synthesis to ribosome synthesis in *Saccharomyces cerevisiae*, J. Biol. Chem. 274(19), 13,235–13,241.

Pollet, N., Schmidt, H., Gawantka, V., Neihrs, C., and Vingron, M. (2000) In silico analysis of gene expression patterns during early development of *Xenopus Laevis*, Proc. Pacific Symp. Biocomput. 5, 440–451.

Shemaker, D. D., Lashkari, D. A., Morris, D., Mittmann, M., and Davis, R. W. (1996) Quantitative phenotypic analysis of yeast deletion mutants using a highly parallel molecular bar-coding strategy, Nat. Genet. 14, 450–456.

Shi, L. (2001) DNA microarray—monitoring the genome on a chip, Technical report, (*see* Website: http://www.gene-chips.com/)

Winzeler, E. A., Shoemaker, D. D., Astromoff, A., Liang, H., Dow, S. W., Friend, S. H. et al. (1999) Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis, Science 285, 901–906.

(2001) ArrayDB, Technical report, National Human Genome Research Institute, Bethesda, MD.
(*see* Website: http://genome.nhgri.nih.gov/arraydb/schema.html)

(2001) ArrayExpress, Technical report, European Bioinformatics Institute, Cambridge, UK.
(*see* Website: http://www.ebi.ac.uk/arrayexpress/index.html)

(2001) Microarray Databases, Technical report, Centre National de la Recherche Scientifique, Paris, France.
(*see* Website: http://www.biologie.ens.fr/en/genetiqu/puces/bddeng.html)

(2001) Microarray Gene Expression Database Group, Technical report, European Bioinformatics Institute, Cambridge, UK.
WWW (*see* Website: http://www.mged.org/)

(2001) Stanford Microarray Database (SMD) Technical report, Stanford University, Stanford, CT.
WWW (*see* Website: http://genome-www5.Stanford.EDU/MicroArray/SMD/)

36 Knowledge Discovery from the Human Transcriptome

Kousaku Okubo and Teruyoshi Hishiki

Introduction

The practical definition of a *transcriptome* is the entire population of mRNAs from a defined source, such as a cell, cells, tissue, or an organism. The population structure, the species of mRNA and their abundance in a transcriptome, varies widely depending on the source. This variation is thought to reflect phenotypic differences between sources. Therefore, the population structure is crucial to understanding the information in the transcriptome data.

The classic analysis of transcriptomes involved liquid hybridization (Rot analysis) in 1970s (*see* Suggested Readings, Lewin [1999]), when most transcripts were anonymous. Nevertheless, important knowledge was extracted regarding a population structure. The population appeared to contain three abundance classes and the number of different transcripts per cell was suggested to be 10,000–20,000.

The situation has changed since the initiation of the human genome project early in the 1990s. Increasing amounts of data regarding the human transcriptome have been generated. In these data, many transcripts are no longer anonymous as they have names or identification codes (IDs), enabling us to compile different data sets and characterize individual genes as well as the transcriptome as a mass.

There are two categories of transcriptome data available in the public domain today. One is population analysis data generated by random sampling identification, and the other is the collection of the ratio of thousands of individual transcripts across different populations. The former is usually carried out by determining short sequences of randomly selected cDNA clones from a library. The latter is generated by microarray hybridization. We will focus on the available random sampling of public data as an example of data structure analysis with the goal of building a framework for such an analysis. Application of the mathematical analysis package, MATLAB (The MathWorks Inc., Natick, MA, *see* Website: http://www.mathworks.com/) that enhances the flexibility in data analysis of transcriptome data will also be discussed.

Data Preparation

To begin mining data for transcriptomes deposited in various forms in the public domain, we must extract population information from the raw data and transform that information into a format that we can apply our statistical analysis and visualization techniques. The desired data format for a transcriptome is a list of component transcript IDs paired with their abundance values in a specific source. If we compile such data from multiple sources, we will have a matrix of abundance values arranged coordinates of transcript ID × source material (Gene Expression Matrix). To simplify the analysis, we do not consider variations in splicing patterns and alternative polyadenylation sites, because discriminative observation is usually not possible. The methods for extracting population data from each type of data are described in the following sections.

EST

An expressed sequence tag (EST) is a short stretch, usually 200–500 bp, of nucleotide sequence from a clone picked randomly from a cDNA library. On one hand the tagged portion usually represents the 3' end of cDNA, because cDNA is usually synthesized with primers annealing to the poly(A) stretch at the 3' end of the mRNA. On the other hand the 5' end of cDNA represents the start site of the transcription. This is only observed when the mRNA remains intact until cDNA synthesis and when it has no strong internal structure that stops reverse transcription. Large numbers of ESTs have been generated (thousands of ESTs per library) from the various projects and deposited into a single public directory, *dbEST*, maintained by the National Center for Biotechnology Information (NCBI). Similar to GenBank, the dbEST data exist as multiple individual nucleic acid sequences with notations of the source material, method for data generation, authors and any citations (for the file formats *see* Websites: http://www.ncbi.nlm.nih.gov/Sitemap/samplerecord.html and http://www.ncbi.nlm.nih.gov/dbEST/). When EST data is used for transcriptome analysis, one first sorts all human EST entries according to their library of origin by using the respective annotations as keys. As shown, some libraries are noted as *normalized*. This means that differences in the representational frequencies between clones were intentionally diminished prior to sampling.

Examples of libraries noted as *normalized*:

NCI_CGAP_Brn41 (Adult brain)
NCI_CGAP_Brn50 (Adult brain)
NCI_CGAP_Brn52 (Adult brain)

You can obtain details about dbEST libraries from the UniGene web pages (*see* Website: http://www.ncbi.nlm.nih.gov/UniGene/lib.cgi?ORG=Hs&LID=816). Library ID (LID)—library source is available from Hs.lib.info, as part of the UniGene files available from the NCBI ftp website.

Other procedures that will alter the representational frequency and population structure include the elimination of abundant clones by prescreening with cDNA probes, and subtraction between two similar libraries to enhance any interpopulation differences. In commercially available libraries, the population of clones may be biased in favor of those that do not interfere with proliferation of host bacteria. ESTs contained within these representationally altered libraries are not suitable for transcriptome analysis. If this information is not noted, we must refer to the original publications to determine if the data are suitable for transcriptome analysis.

To transform the EST records from each library into the population analysis format of transcript ID and abundance, we must cluster ESTs from one library into cognate sequence classes. This process is similar to shotgun sequence assembly using overlap-

Table 1
A Part of the Gene × Tissue Matrix in the BodyMap Database[a]

Blood					*Connective/Muscular*					*Epithelial*					*Cluster ID*		
B01	*B02*	*B03*	*B04*	*B05*	*C01*	*C02*	*C03*	*C04*	*C05*	*E01*	*E02*	*E03*	*E04*	*E05*	*BodyMap*	*UniGene*	*Gene name*
											55				GS08025		cytokeratin 12
											2	23			GS06283	Hs.74070	cytokeratin 13
					3		3	2							GS03142	Hs.195851	a-actin, vascular
	14	1	4	3	2	1	2	11		4	2	2	10	7	GS00244	Hs.180952	b-actin
1	5	1		4	4	1	3	9	9	2	2		4	9	GS00114	Hs.215747	g-actin
					1		11	4							GS02049	Hs.172928	COL1A1
				2			26	13							GS02285	Hs.179573	COL1A2

[a] B01 through E05 represents different libraries.

ping sequences. The results are contained in the UniGene database. In this database, all EST entries and full-length cDNA entries for major organisms have been clustered into cognate sequence classes with unique class IDs (e.g., Hs numbers for human). The correspondence between the UniGene IDs and EST IDs is available (for human *see* Website: ftp://ncbi.nlm.nih.gov/repository/UniGene/Hs.data.Z). Thus, beginning with the EST IDs for a library, we can generate a list of gene IDs with frequency values. Because UniGene has sequence variants registered separately, some UniGene IDs represent the same gene. LocusLink maps UniGene entries to unique gene loci and can be used to integrate multiple UniGene IDs for one gene.

BodyMap

The BodyMap database (*see* Website: http://bodymap.ims.u-tokyo.ac.jp/) is an EST database containing a data set for the purpose of population analysis. The libraries analyzed in the process of building the database are not normalized or amplified. All the ESTs represent the most 3'cDNA fragments created by cleavage with the restriction enzyme *Mbo*I (GATC). The data in BodyMap are summarized as a gene × tissue abundance matrix that shows how frequently a particular gene sequence is represented in a specific source. Table 1 shows a part of such a gene expression matrix that covers approx 20,000 genes × 60 tissues. In BodyMap, the corresponding gene IDs and UniGene IDs are also provided.

SAGE

The serial analysis of gene expression (SAGE) database (*see* Website: http://www.ncbi.nlm.nih.gov/SAGE/) is a collection of very short site-directed tags from cDNA libraries. The sequence tags consist of the 10 bases that follow the 3'-most GATC (MboI) or CATG (NlaIII) restriction site of a cDNAs. Tags are read by sequencing plasmid inserts containing 10–20 concatenated tags to enhance the efficiency of tag collection. The resultant sequences are separated into 10-base tags by separator sequences that are introduced before concatenation. The 10-base tags are counted using a 100% match basis because any 10 bases could represent a gene. If the experiments are carried out as planned and the sequences after the cleavage sites are random, the 10 bases should be long enough to discriminate among the 30,000–50,000+ human genes.

The publicly available SAGE data (*see* Website: ftp://ftp.ncbi.nlm.nih.gov/pub/sage/) is stored in multiple files, each containing a list of 10-base tags with their frequencies in a specific transcriptome. The corresponding Hs number for each tag is provided in the same directory. Because the tags are counted on a perfect-match basis, a sequence alignment program is not necessary, and a simple **sort** command followed by **uniq -c** provided by the UNIX shell is sufficient. Tag frequency and tag-Hs correspondence tables can be used to create a gene × source matrix.

Sources of Confusions and Errors in Data

Random Fluctuation Errors

The difficulty with interpreting the data arises from the random fluctuations or chance errors that are inherent in tag counting. Because the number of clones for a gene is small relative to the size of the sample, the probability to observe x clones for one gene transcript, $p(x)$, should follow the Poisson distribution:

$$p(x) = exp(-\lambda) \cdot \lambda^{x}/x! \qquad \text{[Eq. 1]}$$

where λ is the actual (though unknown) number of clones for a gene per N sampled clones. Figure 1 shows a series of plots of $p(x)$ for $x = 0$ to 10, with λ increasing from 1 to 15. The MATLAB **poisspdf** command has been used. Each curve shows the distribution of the probability to observe x counts for each λ value. For example, given $x = 3$, there could be several values for λ with probability larger than 5%.

The confidence limits for λ, for the observation of x clones of a gene may be obtained as follows. Let α be a significance level (e.g., 0.01), and the lower $1 - \alpha$ confidence limit is

$$L_1 = \chi^2 \{[1 - (\alpha/2)], 2 \cdot x\}/2, \qquad \text{[Eq. 2]}$$

where $\chi\ (p, \nu)$ is the chi-square value for the probability p and the degree of freedom ν, and the upper $1 - \alpha$ confidence limit is

$$L_2 = \chi^2 [\alpha/2, 2 \cdot (x + 1)]/2. \qquad \text{[Eq. 3]}$$

Chi-square values from standard statistical tables for critical values of the chi-square distribution, the Excel function **CHIINV(*p*, ν)** or the MATLAB function **chi2inv(*p*',ν)** (you should use **1 –** *p* for *p*'), can be used. If $x = 3$ and $\alpha = 0.01$, then

$$L_1 = chi2inv\{1 -[1 - (0.01/2)],2 \cdot 3\}/2 = 0.3379,$$

and

$$L_2 = chi2inv[1 - 0.01/2,2 \cdot (3 + 1)]/2 = 10.9775.$$

Therefore, rounding the values yields a 99% confidence interval between 0 and 10 for the count. The expected ranges of actual counts for different observed counts for $\alpha = 0.01$ and 0.05, and the results are presented in Table 2.

Confusion in Tag Counting and in Tag to Gene Correspondence

Cluster errors must also be considered because we count the number of unique tags by clustering the alignments when we use EST data. Errors in clustering have various causes. First, tag sequences containing repetitive sequences may lead to joining different clusters. In addition to repetitive sequences such as Alu or L1, contaminating vector sequences may similiarly act as repetitive units. An example of the inclusion of a repetitive sequence in a representative cDNA set is shown in Table 3. Therefore, sequences are clustered after intensive filtering of repetitive sequences. Nevertheless, there may still be some unmasked repetitive sequence left.

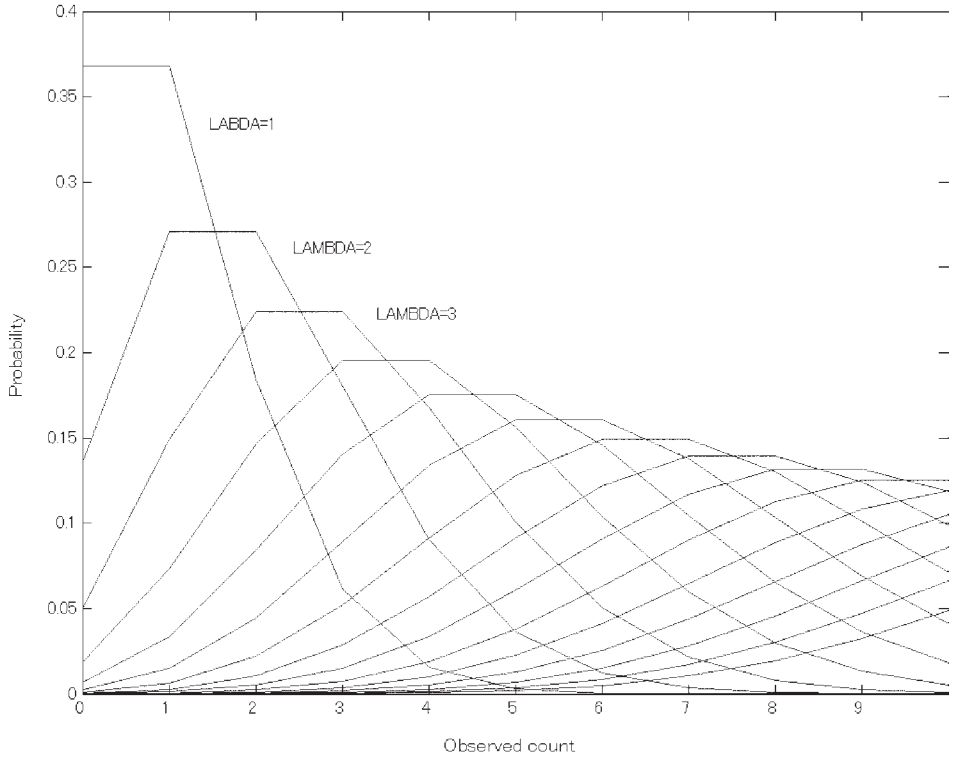


Fig. 1. Possible observations under Poisson distribution.

Table 2
Confidence Intervals of Possible Actual Occurrence (___) for Observed Counts (*x*).

Observed counts		*1*	*2*	*3*	*4*	*5*	*6*	*7*
Actual occurence (λ)	α = 0.05	0–5	0–7	0–8	2–10	2–11	2–13	3–14
	α = 0.01	0–7	0–9	0–10	0–12	2–14	2–15	3–17
8	9	10	11	12	13	14	15	16
4–15	5–17	5–18	6–19	7–20	7–22	8–23	9–24	10–25
3–18	4–19	4–21	5–22	5–24	6–25	7–26	7–28	8–29
17	18	19	20	21	22	23	24	25
10–27	11–28	12–29	13–30	13–32	14–33	15–34	16–35	17–36
9–30	9–32	10–33	11–34	12–35	12–37	13–38	14–39	14–41

It must be remembered that exhaustive attempts to remove repetitive sequences with long margins may lead to loss of unique tags.

Second, clustering errors can be caused by chimeric clones resulting from the connection of two independent cDNA fragments at the cloning steps. Chimeric clones can only be identified when an assembly of cognate tags or an alignment to genomic sequence is made.

The third source of error is contamination by genomic fragments. The canonical polyadenylation signal (AAUAAA) does not assure distinction from genomic

Table 3
Inclusion of a Repetitive Sequence in a Representative cDNA Set

```
A representative sequence containing a repetitive sequence (GS5406):
GANNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNAAAGAAGAAAAAAAGAGAAAAATAACTCTTTT
GAACAAACAGACAAATTAGCTAGTAGTATGGAGATGTATACCCTCTATTACACACATAAA
ACCGTAACAAAATTCATTGTGGTGTATTATAATTAGTTTTGTGAATAGAAAAATAAAGCA
CTTATGTTTAAATTTGTTACAGTTGACTTTTAAAGGATAATGTTGAATCACATTGTCAGA
ATTTTTTCCNCCCCCCGTTCAATTTTGTTNTTTTTACCCCCNAAAATGAAATTCNCAAAT
TTATACNTTTTNTTTTGTTTGAANAATNATCCNCCCNTTTAAAAAAA
The masked part includes a repetitive sequence as follows:
TCGGGTTGAGCCCAGGAGGTTGTGGCTGCAGTNAGCTGTGATGTGCCCTTATGCTATAGCCTGGGCA
AGAGCGTGAGACCCTGTCTCAAAGAAGAAAA
```

sequences. Approximately 10% of the transcripts completely lack this hexanucleotide even if one-base is substituted. A stretch of Ts in 3' ESTs representing poly (A)s may help distinguish such cDNAs; however, when 3' EST sequences are read beyond the poly (A) tail in the antisense direction, the sequences in such a region are often of poor quality and are therefore eliminated from the submitted sequences.

The fourth source of errors is the misidentification of SAGE tags. Sequencing errors are difficult to detect because tags for different genes may differ at only one base. If the sequence quality is 99.0%, the probability of obtaining a 10-bp sequence free from error is 90%. Therefore, there is one incorrect tag in every 10 tags. For example, for 286,089 human SAGE tags, reliable correspondence to an Hs member was found for less than one third (89,230). The remaining tags could be derived from a different region of a partially registered transcript, a novel transcript, or an artifact. This is an excellent example of the inevitable trade-off between efficiency and robustness of random noise in the information transmission.

In SAGE and BodyMap, transcripts without *Mbo*I or *Nla*III cleavage sites are not represented in the data. For example, among 13,739 sequences in the RefSeq database, 398 full-length cDNAs lack *Mbo*I sites and 145 sequences lack *Nla*III sites.

Analyzing Data for a Single Transcriptome

One of the most commonly asked questions about a population of mRNAs is: what are its characteristics? Typical answers have included information such as: the most abundant is gene A, which represents 10% of total tags, followed by B representing 5%; or the 30 most abundant genes comprise 50% of the population; or 500 gene species were found in 1000 tags. These are informative values, but they are not as comprehensive as the classical Rot curves because each value only represents a small fraction of the characteristics of that population.

By introducing a secondary feature to each gene, the abundance rank, we can graphically represent the entire population. The abundance rank is the relative amount of a transcript in a transcriptome, i.e., rank = 1 for the most abundant, rank = 2 for the second most abundant, and so on. The rank is not an inherent feature of a gene, however, it provides a unique description of a gene in a population.

To obtain a graph called the *gene expression profile* the frequencies of genes can be plotted as a function of their abundance ranks. A *virtual Rot curve* can be derived

See companion CD for color Fig. 2

by plotting the accumulated sum of the frequencies of the sequence tags in a population as a function of their frequency rank. These curves look similar for all transcriptomes because the plotted genes are always distributed in small areas close to the coordinates on a logarithmic scale (*see* Fig. 2). These log-rank vs log-frequency plots, or Zipf's plots, were named after a linguist George Kingsley Zipf, who found a beautiful constraint in the frequency of words in English texts. In either case, the continuity of abundance across ranks, rather than clustering to form abundance classes, was suggested. This permits us to easily visualize the difference in mRNA populations between various transcriptomes.

Interestingly, with populations of differentiated and homogeneous cells, such as liver and muscle, the curve is very close to the line representing $F = 0.1/r$, where F is the frequency and the r is the abundance rank. If this holds true for transcripts with the lowest rank, an estimate of the number of transcribed genes in liver or muscle is obtained as a natural number N such that the accumulated sum of $0.1/r$ with r from 1 to N equals 1. For $N = 12{,}367$, the sum equals 1.0000. Although the value may easily be affected by attitude of rare transcripts (tail of the curve), such a prediction is the beauty of the discovery of the constraint.

Useful Tools for Expression Data Analysis of Microarray Data

A set of expression data is essentially a gene by source (e.g., cells and tissues) matrix of frequency data. Therefore, tools to manipulate large matrices efficiently facilitate flexible exploration, rather than having tools specialized in expression profiling. MATLAB, a widely used mathematical computing tool in science and engineering, it is one of the best of its kind. The first version of MATLAB was written in the late 1970s at the University of New Mexico and Stanford University and was intended for use in courses in Matrix Theory. Version 6 (Release 12) contains collections of tools (toolboxes) built on the MATLAB environment, including Statistics Toolbox and Signal Processing Toolbox, have been developed. Student versions are sold at a very reasonable price and provide essentially the same environment, but limit the matrix size.

What is MATLAB?

MATLAB is a software package (*see* Website: http://www.mathworks.com/search) that can be used to define and solve mathematical problems. Its application areas include all fields where problems are defined mathematically. MATLAB provides the flexibility equivalent to problem specific environments written in languages such as Fortran or C. Compatibility with these languages and coverage of computing tasks from data acquisition and analysis to applications development minimizes the learning curve for applying this tool to gene-expression analysis.

MATLAB integrates mathematical computing, visualization, and a powerful technical language. It provides users with a combination of a very intuitive interface, language, and built-in math and graphics functions. For example, let **A** and **B** represent matrices $m \times n$ and $n \times k$. The product of the two matrices is obtained by simply writing the mathematical expression **A * B**. To join the rows (horizontal concatenation) of matrices **A** and **B** to make a wider matrix enter **[A B]**. To display a colored map that represents the distribution of values over the whole matrix *A* requires only that the user run a command **pcolor(A)**.

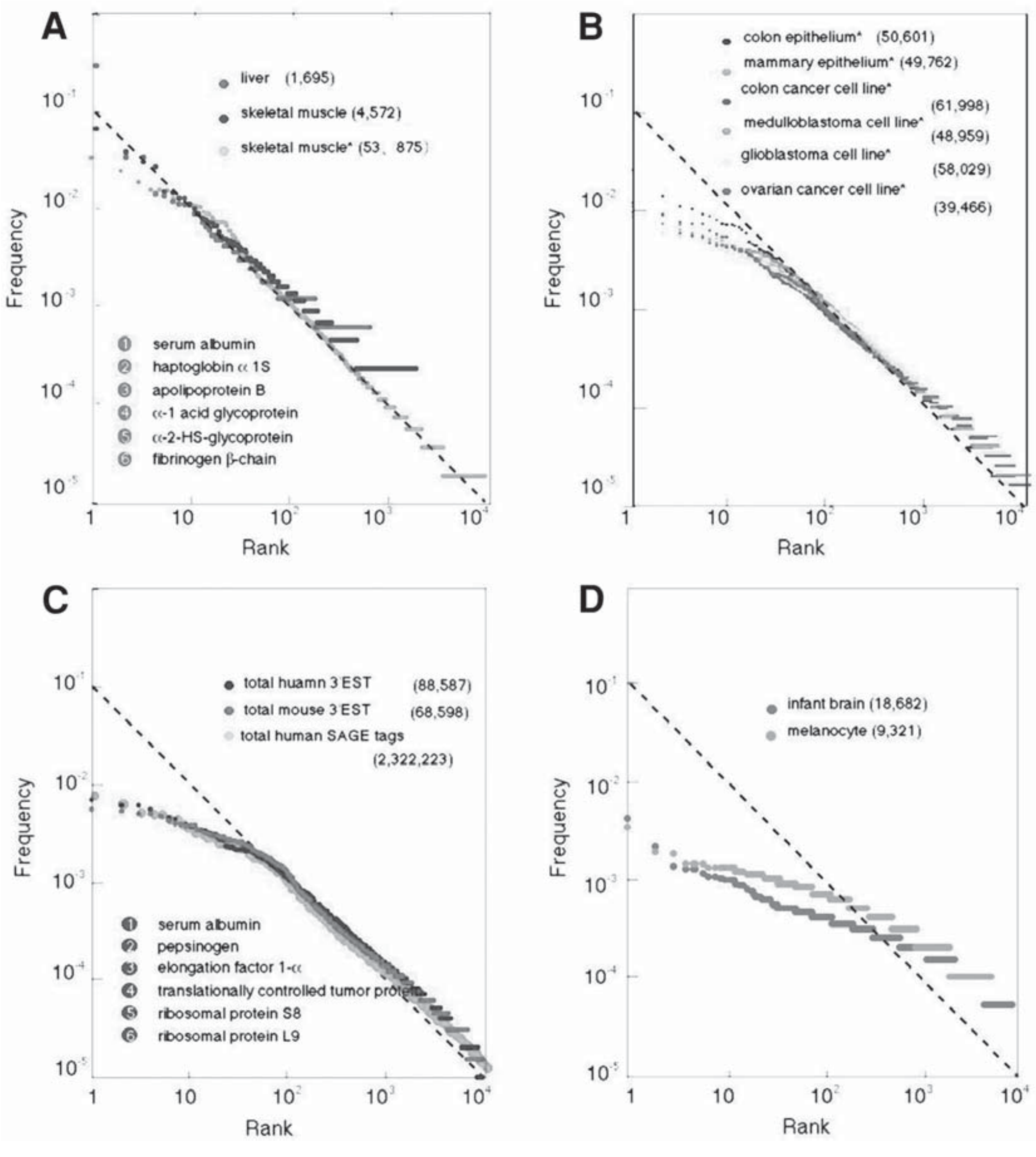


Fig. 2. The frequency of occurrence (*f*) of each transcript in 3'EST and SAGE tag (*) collections of various transcriptomes plotted against the abundance rank (*r*). The broken line represents *f* = 0.1/*r*. **(A)** Organs with homogeneous populations of differentiated cells. Gene names for *r* = 1 – 6 in liver are given. **(B)** Cell lines and complex tissues. **(C)** Compiled data from 51 human EST sets, 31 mouse EST sets, and 64 SAGE tag sets. Gene names for *r* = 1 – 6 in compiled human transcriptome (3'EST) are given. **(D)** Occurrence of 3'ESTs in normalized libraries. The total tag occurrence for each data set is given in parentheses. The frequency data were obtained from websites (*see* Websites: http://bodymap.ims.u-tokyo.ac.jp/datasets/index.html (3'EST) and ftp://ncbi.nlm.nih.gov/pub/sage/ (SAGE). The data for liver are the combined data for two human liver libraries. The frequencies of total SAGE tags are from re-analysis of all available human SAGE tags. Clustering 3'ESTs for two representative normalized libraries in dbEST, 1N1B and 2NbHM, yielded the data for normalized libraries.

Expression Analysis with MATLAB

This section provides an example of the use of MATLAB to determine various expression patterns. In this example, we will use supplemental data (Fig. 2 clusters; **fig2data.txt**) that were originally reported in Iyer et al. (1999). This can be downloaded from Stanford Genomic Resource (*see* Website: http://genome-www.stanford.edu/).

Import and Display Matrix Data

First, import and display an entire matrix. The raw matrix data should be a text delimited file with tabs or other delimiters (e.g., spaces and |). The command **help formats** shows other compatible file formats. Pull down the **File** menu, start the Import Wizards and load **fig2data.txt** The file **fig2data.txt** has several comment lines that provide an explanation of the origin of the file. These lines should be removed. Some text files are impossible to import because of format errors, that are often hard to detect. In such a case, importing these files into Excel and exporting the files as a tab-delimited text file is often effective. The wizards will separate the text file into a data matrix named *data* and row/column titles named *textdata*.

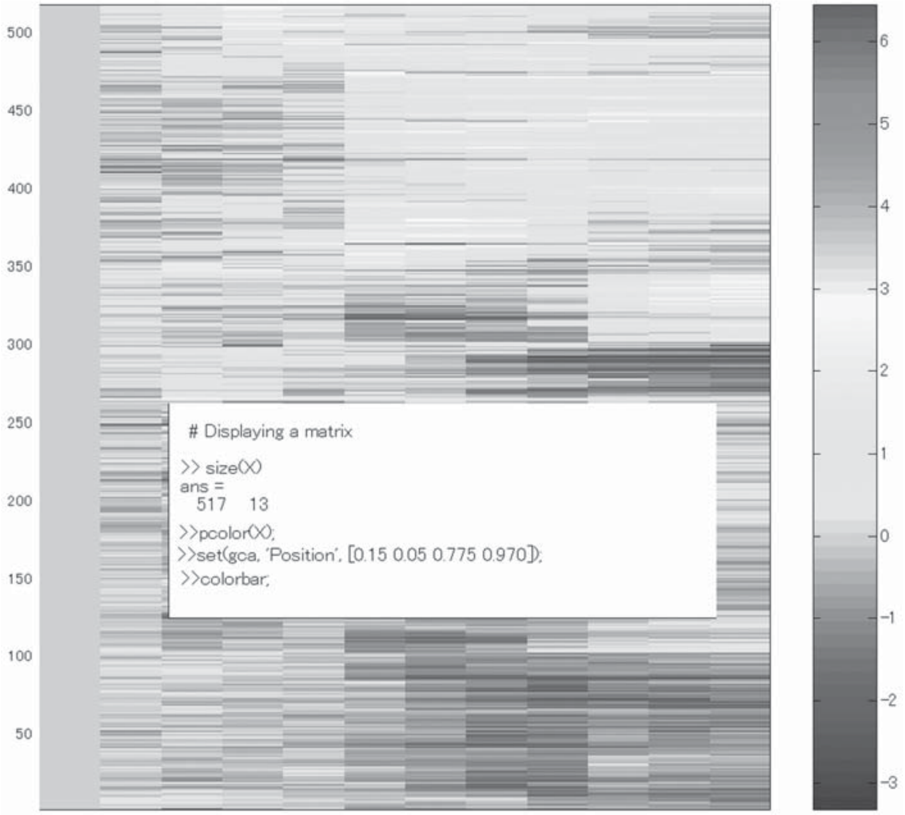
The command **pcolor(data)** will display the matrix. Often only a black window is visible because the panel is too small to both draw mesh lines and paint matrix cells. Pull down the **Edit** menu, start the Current Object Property Editor, and remove the mesh lines by setting the line style to **none**. The color-value table (color bar) for the matrix cells is shown as the **colorbar** command. Only a blue field will be visible as the values in the cells are the red/green ratio scanned from the microarrays. The lower limit of the ratio is 0 while the upper limit may be more than 100. It is necessary to use a logarithmic scale as most matrix cells are in the blue (near 1) area. Let Fig2 represent a new data matrix, and make a statement **Fig2 = log2(data)** to make a properly scaled matrix. Figure 3 shows the column 1 (0 h) to 13 (unsync) of Fig. 2 in Iya et al. (1999).

Let *X* be a matrix, which has been extracted from *Fig2* by **X = Fig2(:,1:13);**. Run **pcolor(X)** to display the figure. To reverse the direction of the Y-axis, type **set(gca,'YDir', 'reverse')**. The direction can be changed again by **set(gca, 'YDir', normal')**. The background color of the figure can be changed to white using the **set gcf,'color','white'** command.

Manipulating the Color Map

Readers may want to change the color bar to the familiar red/green representation, because it is rational to use only two colors rather than to use all three (red, green, and blue) to show the ratio between two mRNA levels. This is possible by changing the color map or a table that relates the order in the value to a color. A color map is a matrix of three columns, where each row represents a color, and the first, second, and third columns represent the intensities of the red, green, and blue components, respectively.

We will make a color map with the same number of rows, 64, as the default, but it will have a red column with a continuum of zeros from the start to the middle and then increase linearly from 0 to 1. The color map will have a green column that starts at 1 and diminishes from 1 to 0 in the middle, with a continuum of zeros to the end, and a blue column with all zeros. Lower values correspond to the intensity of green, whereas higher values correspond to the intensity of red. The color map has a black



```
# Displaying a matrix

>> size(X)
ans =
  517   13
>>pcolor(X);
>>set(gca, 'Position', [0.15 0.05 0.775 0.970]);
>>colorbar;
```

Fig. 3. Displaying a matrix.

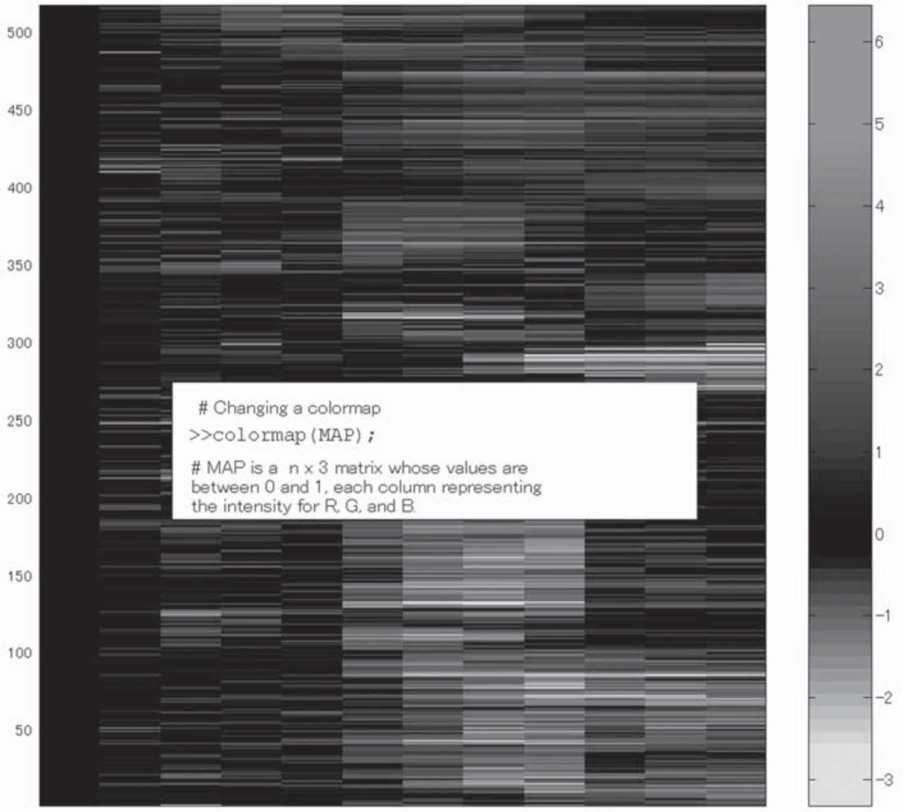
row with all zeros in the middle, and this can be adjusted so that cells with red/green ratios close to 1.0 are black. Name the color matrix **MAP**. To use the color map, display the figure and then execute **colormap(MAP)**. Figure 4 shows the resultant representation.

Preparation Before Clustering

MATLAB can be used to cluster by rebuilding a clustered matrix of expression patterns. In this section, a random permutation (ordering) of the rows is generated from the original clustered matrix. To start, make a random permutation array of length 517 or the number of rows. A permutation array is a reordered list of serial numbers indicating the order of the numbered objects. If the *i*-th element of a permutation array is *j*, the *j*-th object occupies the *i*-th position in the new list. Let the array **p_1 = randperm(517)**. With this array, a row permutation matrix that reorders the rows of a matrix can be simply defined by **X_rowrand = X(p_1,:)**. An example of a matrix with permutated rows of X is shown in Fig. 5.

Clustering Expression Data

There are two purposes to clustering the data: 1) to search for a set of genes showing similar expression patterns, and 2) to search for similar cells or tissues showing



```
# Changing a colormap
>>colormap(MAP);
# MAP is a  n x 3 matrix whose values are
between 0 and 1, each column representing
the intensity for R, G, and B.
```

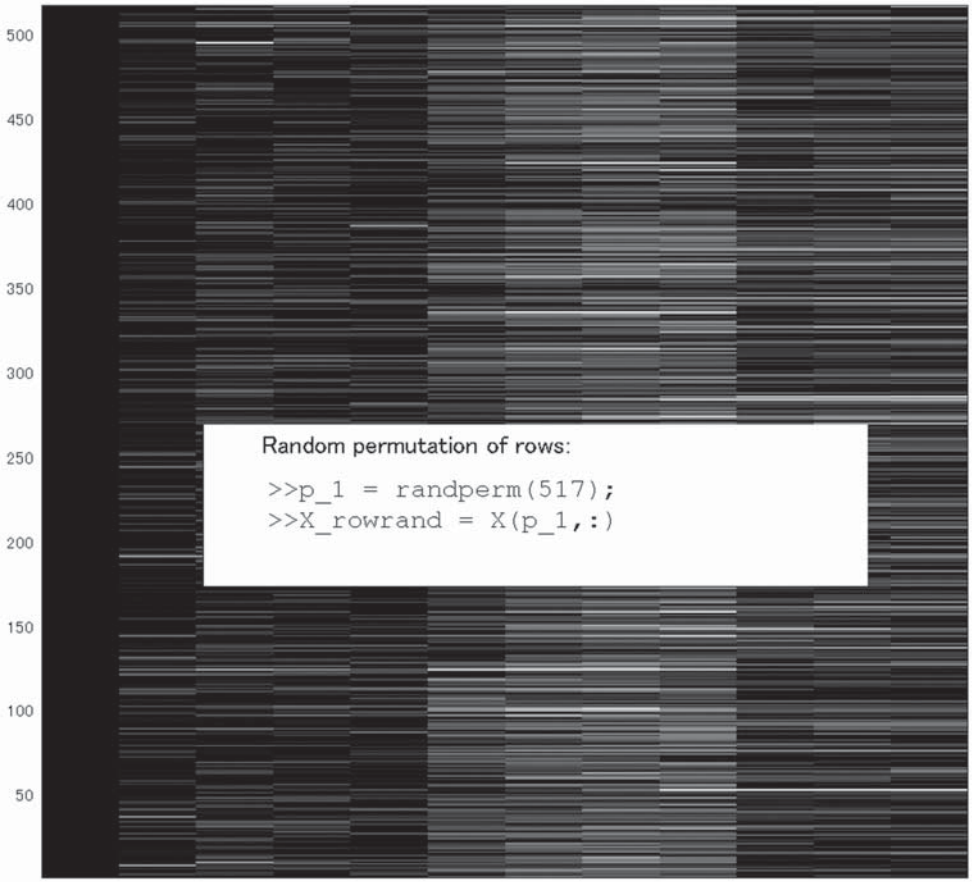
Fig. 4. Changing the color map.

similar patterns of expression. The first is referred to as *row clustering*, and the second *column clustering*.

This example has time-ordered, noncategorical columns and therefore, only row clustering will be considered. Column clustering can be done independently and is accomplished the same way as row clustering after the matrix is transposed. The MATLAB expression of the transposed matrix A is *A*.

One of the methods to cluster the rows is hierarchical clustering. This is accomplished in three steps: 1) calculating pairwise distances between data points (row vectors); 2) linking the nearest data point or group of data points hierarchically (to make a tree) with some alternative methods of defining the distance between groups; 3) cutting the tree into clusters according to a dissimilarity threshold within a cluster or the desirable number of clusters.

MATLAB users can easily define these tasks and make tests. For example, the pairwise distance calculation task is carried out with Euclidian distance by **Y = pdist(X_rowrand, 'euclid')**, and the linkage and tree generation task with average distance between data points in two groups as the distance between the groups by **Z = linkage(Y, 'average')**. If the dissimilarity threshold is set to 0.95, cutting the tree or the clustering is **T = cluster(Z, '0.95')**. The array *T* has a length of the number of rows of *X*, and the *i*-th element corresponds to the *i*-th row of *X_rowrand*; the value of the



Random permutation of rows:

```
>>p_1 = randperm(517);
>>X_rowrand = X(p_1,:)
```

Fig. 5. A randomly row-permutated matrix.

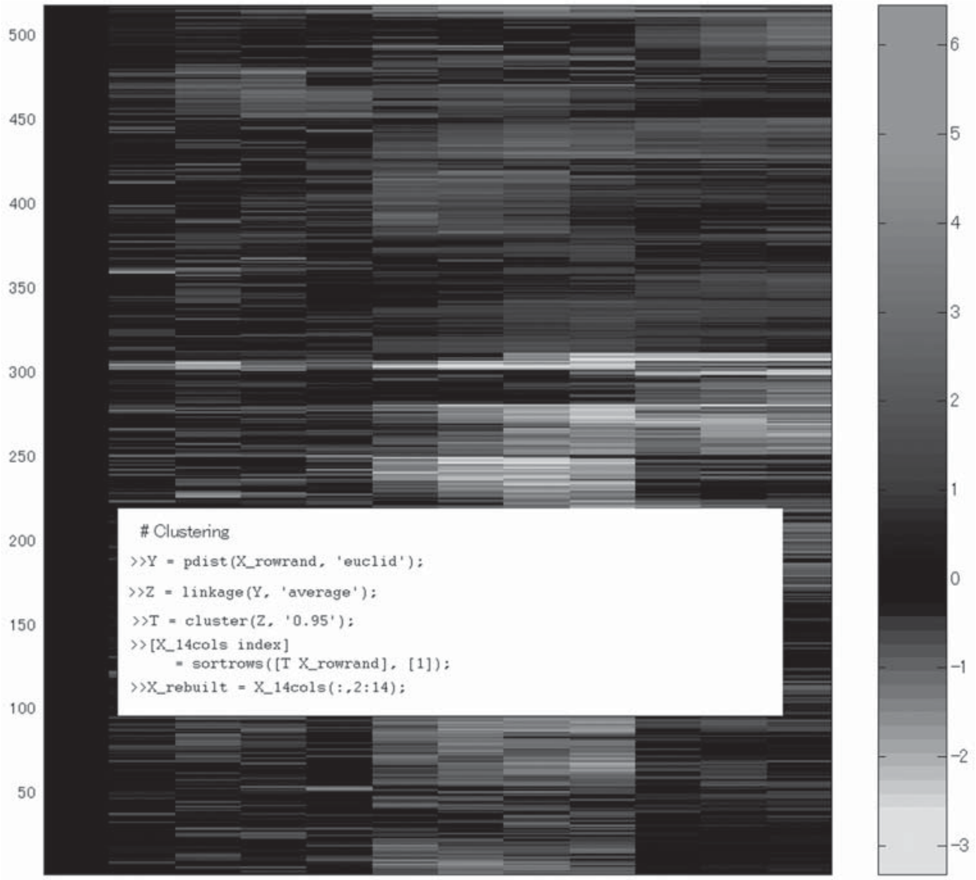
element is the cluster to which the corresponding row is assigned. The number of clusters is obtained by **max(T)**.

Sort the randomized rows by the cluster array *T* with **[X_14cols index]= sortrows([T X_rowrand], [1]);**. The resultant clustered matrix is obtained by **X_rebuilt = X_14cols(:,2:14);**. The **sortrows** command sorts the rows of a matrix **[T X_rowrand]** by the columns indicated by a list **[1]**. The matrix **X_rebuilt** is shown as Fig. 6. The **index** array directs the permutation of rows of the randomized matrix, **X_rowrand**.

See companion CD for color Fig. 6

To see how similar the reclustered matrix is to the original, a matrix describing the correspondence of rows between the matrices can be generated by **P2 = spconvert([(1:517)' p_1(index)' ones(517,1)]);**. The command **spy(P2);** plots the matrix as shown in Fig. 7.

The closer the points are to the diagonal line that run from the upper left to the lower right of the plot, the greater the similarity between matrices. The figure shows that the positions of some clusters are interchanged each other but the set of clusters as a whole is preserved after reclustering.



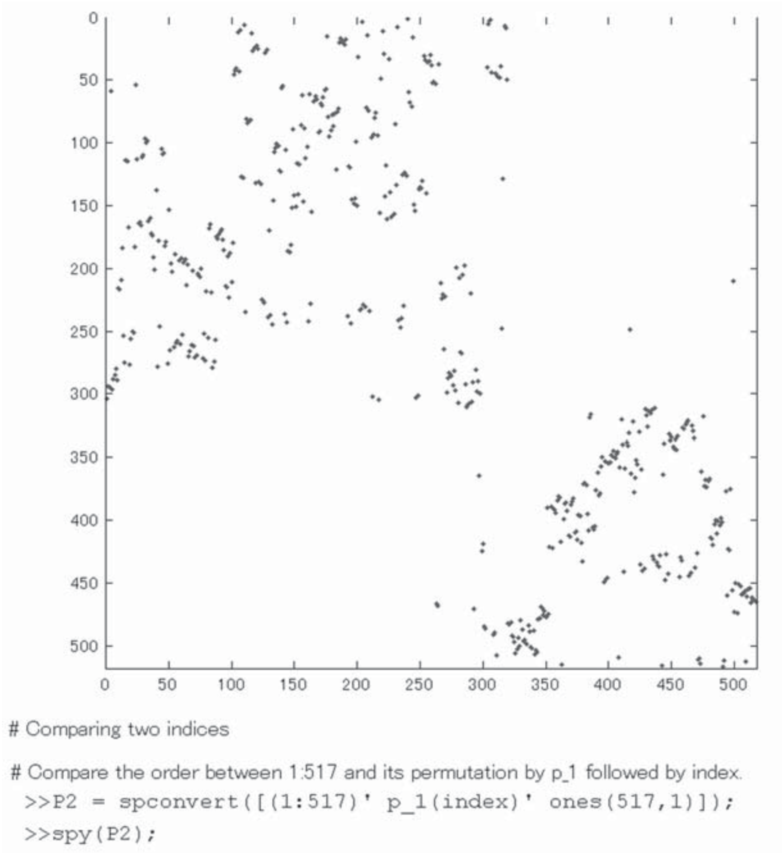
```
# Clustering
>>Y = pdist(X_rowrand, 'euclid');
>>Z = linkage(Y, 'average');
>>T = cluster(Z, '0.95');
>>[X_14cols index]
     = sortrows([T X_rowrand], [1]);
>>X_rebuilt = X_14cols(:,2:14);
```

Fig. 6. The re-clustered matrix.

Displaying a Part of the Matrix

To examine the expression matrix more closely, extract a part of the matrix and attach titles to the rows. For example, extract rows 271–320 from the rebuilt matrix and rows 251–300 from the original matrix. Correspondence between these rows is indicated in Fig. 7. Display the rebuilt matrix with **pcolor(X_rebuilt(271:320,:))**. However, before running that command the value range to adjust the color map must be defined.

By default, the MATLAB color map allocates the smallest value to the first row of the color map matrix and the largest to the last row without considering the absolute value. Therefore, when a small part of the matrix is displayed, the range of the value in the whole data must be explicitly defined. Display the original matrix *X* by **pcolor(X)**. Run the **cax = caxis** command. If a semicolon is not included at the end of the statement, the command will show the content of the *cax* array, which is the range of the values of the cells in the whole matrix. Now display the partial matrix and run **colormap(MAP)**. The colors will change. Type **caxis(cax)** to adjust the range of colors. To include more areas of the matrix, run a command such as **set(gca, 'Position', [0.15 0.05 0.775 0.970]);**.

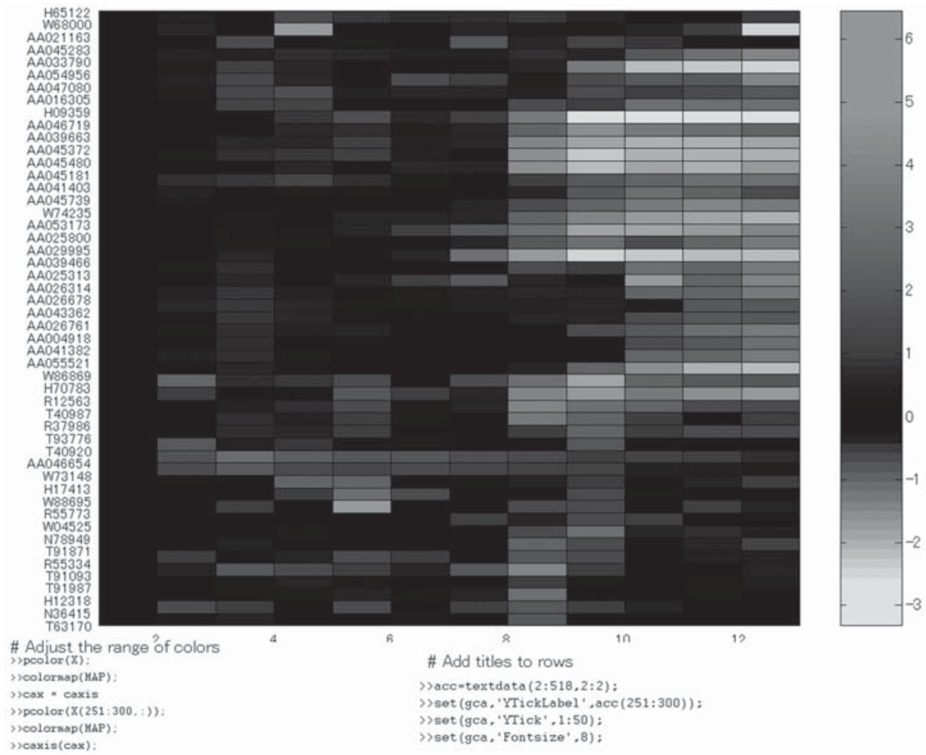


```
# Comparing two indices

# Compare the order between 1:517 and its permutation by p_1 followed by index.
  >>P2 = spconvert([(1:517)' p_1(index)' ones(517,1)]);
  >>spy(P2);
```

Fig. 7. Comparison of two indices.

To assign titles to each row, make titles from the *textdata* array that was separated when importing. To obtain an array of GenBank EST accession numbers for the genes, extract the second column as **acc = textdata(2:518,2:2)**”. A version that has gone through randomization and clustering, *acc_rebuilt*, can be obtained with **acc_rebuilt = acc(p_1(index))**. The labels and their locations on the Y-axis can be changed. Define a label set and the order of labels by **set(gca, 'YtickLabel', acc_rebuilt(271:320))**. Set the number of labels to 50 by **set(gca, 'Ytick', 1:50)**. The font size can be changed to 8 points with **set(gca,'Fontsize',8)**. Now display the color bar by **colorbar**. Compare Fig. 8, which shows the original matrix with Fig. 9, which shows the rebuilt matrix.

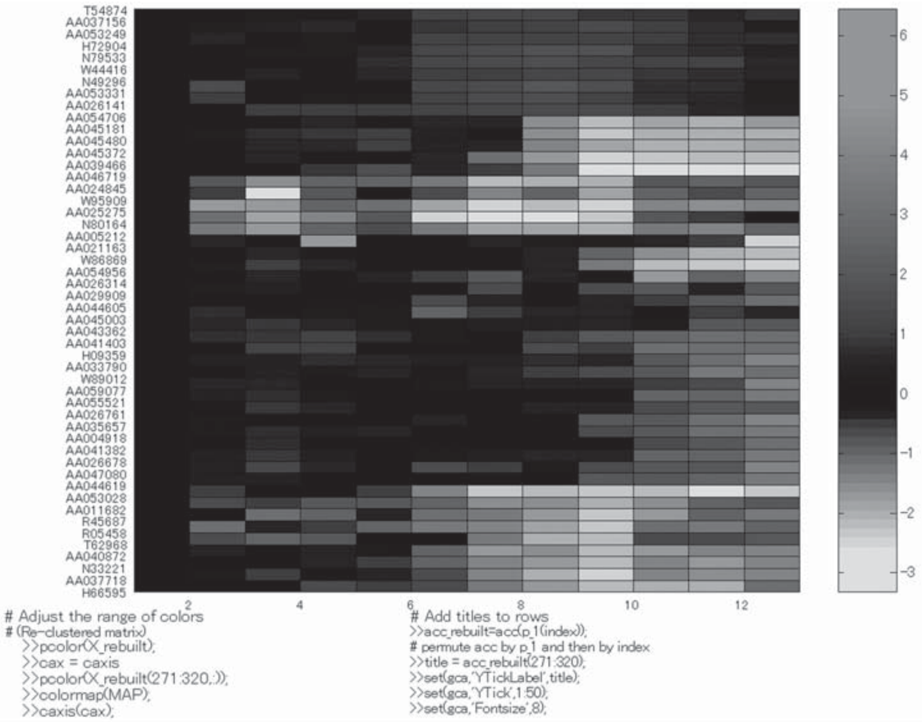
Rows can also be extracted by cluster. As an example, rows that belong to clusters whose serial numbers are larger than 150 will be extracted. Use the array *T* that relates rows of the randomized matrix *X_rowrand* to the cluster serial number as the result of **cluster** function. Let an array *T_sort* represent row-to-cluster correspondence in *X_rebuilt*. It is obtained by **T_sort = sort(T);**. An array of the row indices for clusters with a serial number larger than 150 is obtained by **find(T_sort > 150);**. With this array, rows are extracted by **X_rebuilt(find(T_sort > 150),:)**, and the titles are extracted by **acc_rebuilt(find(T_sort > 150));**.



```
# Adjust the range of colors
>>pcolor(X);
>>colormap(MAP);
>>cax = caxis
>>pcolor(X(251:300,:));
>>colormap(MAP);
>>caxis(cax);
```

```
# Add titles to rows
>>acc=textdata(2:518,2:2);
>>set(gca,'YTickLabel',acc(251:300));
>>set(gca,'YTick',1:50);
>>set(gca,'Fontsize',8);
```

Fig. 8. A portion of the original matrix.



```
# Adjust the range of colors
# (Re-clustered matrix)
   >>pcolor(X_rebuilt);
   >>cax = caxis
   >>pcolor(X_rebuilt(271:320,:));
   >>colormap(MAP);
   >>caxis(cax);
```

```
# Add titles to rows
>>acc_rebuilt=acc(p_1(index));
# permute acc by p_1 and then by index
>>title = acc_rebuilt(271:320);
>>set(gca,'YTickLabel',title);
>>set(gca,'YTick',1:50);
>>set(gca,'Fontsize',8);
```

Fig. 9. The corresponding portion of the re-clustered matrix.

Analyzing Data for Multiple Transcriptomes and Future Prospects

Abundance matrices constructed from random sampling data are typically of very sparse low sensitivity. For example, matrices constructed from the dbEST and SAGE only contain values in approx 5% of the cells. This makes the random sampling data unsuitable for clustering of genes by source distribution patterns and clustering of sources by mRNA. Nevertheless, the entire matrix is useful for selecting genes on the basis of distribution patterns especially for transcripts with high ranks in some tissues. Absolute abundance data is more reliable than the absolute intensities of hybridization signals because hybridization signals are dependent both on the concentration of labeled cRNA/cDNA molecules and their stereostructure in solution. For these reasons, total abundance levels of transcripts obtained from random sampling data will continue to provide reliable clues for predicting gene functions from expression patterns. Examining correlations between absolute abundance levels and gene characteristics such as GC content and primary transcript length may be worthwhile.

One goal of transcriptome data mining is the systematic annotation of gene functions. Microarray experiments have revealed that transcripts with high similarity in tissue distributions share some functional properties. However, discovering the properties shared by a group of known transcripts is experts' work, and such human efforts cannot keep pace with the rate of production of transcriptome data. Moreover, manual annotations are typically not reproducible or objective and must be repeated because the knowledge available regarding genes increases daily.

To overcome the inconvenience of manual annotation, developing a system for *computing with function* is a central issue of biomedical research. Computing with function is defined as executing computation operations whose inputs or outputs are descriptions of the functions of biomolecules. Efforts for formal representation of biochemical knowledge by KEGG and EcoCyc have pioneered this effort. Gene Ontology (*see* Website: http://www.geneontology.org) also facilitates such computations by introducing controlled and structured terminology for describing gene function. Although such efforts have proven effective for simple organisms, the same task in complex organisms represents ontological challenges where most proteins are embedded in regulatory networks that cannot be represented as lower order structures.

Suggested Readings

Transcript Abundance

Bishop, J. O., Morton, J. G., Rosbash, M., and Richardson, M. (1974) Three abundance classes in HeLa cell messenger RNA, Nature 250, 199–204.

Lewin, B., (1999) Genes VI, Oxford University Press, New York, NY.

Model Transcriptome Analysis

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., et al. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature 403, 503–511.

Gilbert, D. R., Schroeder, M., and Helden, J. (2000) Interactive visualization and exploration of relationships between biological objects, Trends Biotechnol. 18, 487–494.

Iyer, V. R., Eisen, M. B., Ross, D. T., Schuler, G., Moore, T., Lee, J. C., et al. (1999) The transcriptional program in the response of human fibroblasts to serum, Science 283, 83–87.

Okubo, K., Hori, N., Matoba, R., Niiyama, T., Fukushima, A., Kojima, Y., and Matsubara, K. (1992) Large scale cDNA sequencing for analysis of quantitative and qualitative aspects of gene expression, Nat. Genet. 2, 173–179.

Perou, C. M., Jeffrey, S. S., van de Rijn, M., Rees, C. A., Eisen, M. B., Ross, D.T., et al. (1999) Distinctive gene expression patterns in human mammary epithelial cells and breast cancers, Proc. Natl. Acad. Sci. USA. 96, 9212–9217.

Ross, D.T., Scherf, U., Eisen, M. B., Perou, C. M., Rees, C., Spellman, P., et al. (2000) Systematic variation in gene expression patterns in human cancer cell lines, Nat. Genet. 24, 227–235.

Velculescu, V. E., Zhang, L., Vogelstein, B., and Kinzler, K. W. (1995) Serial analysis of gene expression, Science 270, 484–487.

Data Preparation

EST

Adams, M. D., Kelley, J. M., Gocayne, J. D., Dubnick, M., Polymeropoulos, M. H., Xiao, H., et al. (1991) Complementary DNA sequencing, expressed sequence tags and human genome project, Science 252, 1651–1656.

Boguski, M. S., Lowe, T. M., and Tolstoshev, C. M. (1993) dbEST—database for “expressed sequence tags”, Nat. Genet. 4, 332–333.

Pruitt, K. D., Katz, K. S., Sicotte, H., and Maglott, D.R. (2000) Introducing RefSeq and LocusLink, curated human genome resources at the NCBI, Trends Genet. 16, 44–47.

Schuler, G. D., Boguski, M. S., Stewart, E. A., Stein, L. D., Gyapay, G., Rice, K., et al. (1996) A gene map of the human genome, Science 274, 540–546.

Organism Transcriptome Maps

Hishiki, T., Kawamoto, S., Morishita, S., and Okubo, K. (2000) BodyMap: a human and mouse gene expression database, Nucleic Acids Res. 28, 136–138.

Kawamoto, S., Yoshii, J., Mizuno, K., Ito, K., Miyamoto, Y., Ohnishi, T., et al. (2000) BodyMap: a collection of 3' ESTs for analysis of human gene expression information, Genome Res. 10, 1817–1827.

Lash, A.E., Tolstoshev, C. M., Wagner, L., Schuler, G. D., Strausberg, R. L., Riggins, G. J., Altschul, S. F. (2000) SAGEmap: A public gene expression resource, Genome Res. 10, 1051–1060.

Sese, J., Nikaidou, H., Kawamoto, S., Minesaki, Y., Morishita, S., and Okubo, K. (2001) BodyMap incorporated PCR-based expression profiling data and a gene ranking system, Nucleic Acids Res. 29, 156–158.

Microarrays, Expression and Statistical Analysis Techniques

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, The Gene Ontology Consortium, Nat. Genet. 25, 25–29.

Brenner, S. E. (1999) Errors in genome annotation, Trends Genet. 15, 132–133.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA 95, 14,863–14,868.

Kanehisa, M. and Goto, S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes, Nucleic Acids Res. 28, 27–30.

Karp, P. D. (2000) An ontology for biological function based on molecular interactions, Bioinformatics 16, 269–285.

Karp, P. D., Riley, M., Saier, M, Paulsen, I. T., Paley, S. M., and Pellegrini-Toole, A. (2000) The EcoCyc and MetaCyc databases, Nucleic Acids Res. 28, 56–59.

MathWorks, Inc. (1997) The Student Edition of MATLAB, User's Guide, version 5, Prentice-Hall, Upper Saddle River, NJ.

Southern, E., Mir, K., and Shchepinov, M. (1999) Molecular interactions on microarrays, Nat. Genet. 21(1 Suppl), 5–9.

Velculescu, V. E., Madden, S.L., Zhang, L., Lash, A. E., Yu, J., Rago, C., et al. (1999) Analysis of human transcriptomes, Nat. Genet. 23, 387–388.

Zar, J. H., (1999) Biostatistical Analysis, 4th ed., Prentice-Hall International, Upper Saddle River, NJ, p. 574.

Zipf, G. K. (1949) Human behavior and the principle of least effort, Addison-Wesley, Cambridge, MA.

Part V

Appendices

1. **CD Contents**
2. **A Collection of Useful Bioinformatic Tools and Molecular Tables**
3. **Simple UNIX Commands**

1 Appendix

CD Contents

What is Included on the CD?

The CD that comes with this book includes:

1. All Figures and Tables with legends from the various chapters, many of which are in color. This is an excellent source of illustrative material for presentations.
2. Several bioinformatics software packages that the readers can install on their own computer workstations or servers.
3. Several useful basic tables and charts for understanding genome properties.

The CD is organized into folders and subfolders. The readers should be able to load the CD into the CD-drive of any IBM-Personal Computer or Apple Macintosh and browse through the folders.

The color figures can be found in the **Color Figures** folder, organized into subfolders by chapter.

The software packages can be found in the **Programs** folder, organized into subfolders by the name of each package. For each program subfolder, there is a **ReadmeCD** file that provides further information about the software, including how to install it, use it, and where up-to-date versions can be downloaded from the Web. There is also information on licensing and registration, and restrictions that may apply.

BioDiscovery

This folder contains software packages for microarray analysis that may be installed on IBM-PC computers. Installation instructions are included in the file named **Readme.pdf.** You will need to use the Acrobat Reader utility to read the file (*see* Section "Adobe Acrobat Reader"). The BioDiscovery software was kindly provided by Sorin Draghici, author of Chapter 35.

ClustalX

This folder contains the graphical interface versions of the Clustal multiple sequence alignment program. Versions for both IBM-PC (clustalx1.81.msw.zip) and Macintosh (clustalx1.81.PPC.sea.Hqx) are included. The files in the packages will need to be unpacked with common unzipping utilities. ClustalX versions for various flavors of UNIX are also available from the original source FTP website (*see* Website: ftp://ftp-igbmc.u-strasbg.fr/pub/ClustalX/), described in the readme file. Permission to include ClustalX on this CD was kindly provided by Julie Thompson and is described by Steven Thompson in Chapter 31.

Ensembl

This folder contains the files needed to install the Ensembl package on a UNIX server. Installation instructions are located in the **additional docs** subfolder in the file named **EnsemblInstall100.pdf**. You will need to use the Acrobat Reader utility to read the file (*see* Section "Adobe Acrobat Reader"). The **source code** subfolder contains the required source code for both Ensembl and Bioperl. Note that the files in the source code folder are in UNIX format. Please use BINARY FTP mode to transfer those files to your UNIX server. Up-to-date versions of Ensembl and Bioperl are available at their respective Websites (*see* Websites: http://www.ensembl.org/ and http://www.bioperl.org). The Ensembl software was kindly provided by James Stalker, author of Chapter 25.

MicroAnalyser

This folder contains a software package for microarray analysis that may be installed on Macintosh computers. Up-to-date versions of the software are available (*see* Website: http://imru.bham.ac.uk/MicroAnalyser/). Permission to include the MicroAnalyser software on this CD was kindly provided by Adrian Platts.

Oligo

This folder contains demo versions of the *Oligo* primer design and analysis software for both IBM-PC and Macintosh computers. This software was kindly provided by Wojciech Rychlik, author of Chapter 21.

Sequencealign

This folder contains a PowerPoint demonstration of sequence alignment. It was kindly contributed by David S. Wishart, author of chapter 27.

Singh_perl_scripts

This folder contains perl scripts for statistical analysis that were generously contributed by Gautam Singh, author of Chapters 22 and 23. They can be used for solving the problems described in Chapter 23.

Staden

This folder contains the Staden Sequence Analysis Package and the Gap4 Viewer software that can be installed on an IBM-PC computer. For up-to-date versions *see* Website: http://www.mrc-lmb.cam.ac.uk/pubseq/. This software was kindly provided by Roger Staden, author of Chapters 20 and 24.

TreeView

This folder contains the TreeView tree drawing software for both IBM-PC and Macintosh computers. TreeView is a free program for displaying phylogenies. Up-to-date versions, including UNIX versions, can be found (*see* Website: http://taxonomy.zoology.gla.ac.uk/rod/treeview.html). Please visit the Website to register TreeView if you wish to use it. Permission to include TreeView on this CD was kindly provided by Roderic D. M. Page.

Adobe Acrobat Reader

In several of the folders on the CD, there are information files that may be in PDF format. To read PDF format files, you will need to have the free Acrobat Reader utility installed on your computer. If you do not already have Acrobat Reader installed, you can download it (*see* Website:: http://www.adobe.com/products/acrobat/readstep.html).

Other Sources for Bioinformatics Software

There are many sources available for downloading software that may be useful. Here are two of our favorites: EBI FTP Server (*see* Website: http://www.ebi.ac.uk/FTP/) and IUBio Archive for Biology data and software (*see* Website: http://iubio.bio.indiana.edu/).

2 Appendix

A Collection of Useful Bioinformatic Tools and Molecular Tables

The Genetic Code

	2nd Position				
1st Position	**U**	**C**	**A**	**G**	*3rd Position*
U	UUU Phe	UCU Ser	UAU Tyr	UGU Cys	**U**
	UUC Phe	UCC Ser	UAC Tyr	UGC Cys	**C**
	UUA Leu	UCA Ser	**UAA** Stop	**UGA** Stop	**A**
	UUG Leu	UCG Ser	**UAG** Stop	UGG Trp	**G**
C	CUU Leu	CCU Pro	CAU His	CGU Arg	**U**
	CUC Leu	CCC Pro	CAC His	CGC Arg	**C**
	CUA Leu	CCA Pro	CAA Gin	CGA Arg	**A**
	CUG Leu	CCG Pro	CAG Gin	CGG Arg	**G**
A	AUU Ile	ACU Thr	AAU Asn	AGU Ser	**U**
	AUC Ile	ACC Thr	AAC Asn	AGC Ser	**C**
	AUA Ile	ACA Thr	AAA Lys	AGA Arg	**A**
	AUG Met	ACG Thr	AAG Lys	AGG Arg	**G**
G	GUU Val	GCU Ala	GAU Asp	GGU Gly	**U**
	GUC Val	GCC Ala	GAC Asp	GGC Gly	**C**
	GUA Val	GCA Ala	GAA Glu	GGA Gly	**A**
	GUG Val	GCG Ala	GAG Glu	GGG Gly	**G**

The codons are read as triplets in the 5' → 3' direction, i.e., left to right.
Termination codons are in bold.

IUPAC Nucleotide Codes

Code	*Members*	*Nucleotide*
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine (DNA)
U	U	Uracil (RNA)
Y	C or T(U)	pYrimidine
R	A or G	puRine
M	A or C	aMino
K	G or T(U)	Keto
S	G or C	Strong interaction (3 H bonds)
W	A or T(U)	Weak interaction (2 H bonds)
H	A or C or T(U)	not-G
B	G or T(U) or C	not-A
V	G or C or A	not-T
D	G or A or T(U)	not-C
N	G,A,C or T(U)	aNy base

IUPAC Amino Acid Codes

3-Letter Code	*1-Letter Code*	*Amino Acid*
Ala	A	Alanine
Arg	R	Arginine
Asn	N	Asparagine
Asp	D	Aspartic acid
Cys	C	Cysteine
Gln	Q	Glutamine
Glu	E	Glutamic acid
Gly	G	Glycine
His	H	Histidine
Ile	I	Isoleucine
Leu	L	Leucine
Lys	K	Lysine
Met	M	Methionine
Phe	F	Phenylalanine
Pro	P	Proline
Ser	S	Serine
Thr	T	Threonine
Trp	W	Tryptophan
Tyr	Y	Tyrosine
Val	V	Valine
Asx	B	Aspartic acid or Asparagine
Glx	Z	Glutamic acid or Glutamine
Xaa	X	Any amino acid

Converting Base Size of a Nucleic Acid → Mass of Nucleic Acid

Number of Bases	*Mass of Nucleic Acid*
1 kb ds DNA (Na$^+$)	6.6 × 10^5 Da
1 kb ss DNA (Na$^+$)	3.3 × 10^5 Da
1 kb ss RNA (Na$^+$)	3.4 × 10^5 Da
1.52 kb ds DNA	1MDa ds DNA (Na$^+$)
Average MW of a dsDNA	660 Da
Average MW of a ss DNA	330 Da
Average NW of an RNA	340 Da

Converting Base Size of a Nucleic Acid → Maximum Moles of Protein

DNA	*Molecular Weight (Da)*	*Amino Acids*	*1 μg*	*1 nmol*
270 bp	10,000	90	100 pmol or 6 × 10^{13} molecules	10 μg
1.35 Kbp	50,000	450	20 pmol or 1.2 × 10^{13} molecules	50 μg
2.7 Kbp	100,000	900	10 pmol or 6 × 10^{12} molecules	100 μg
4.05 Kbp	150,000	1350	6.7 pmol or 4 × 10^{12} molecules	150 μg

Average MW of an amino acid = 110 (Da).
3 bp are required to encode 1 amino acid.

Sizes of Common Nucleic Acids

Nucleic Acid	*Number of Nucleotides*	*Molecular Weight*
lambda DNA	48,502 (dsDNA)	3.2 × 10^7
pBR322 DNA	4361 (dsDNA)	2.8 × 10^6
28S rRNA	4800	1.6 × 10^6
23S rRNA (*E. coli*)	2900	1.0 × 10^6
18S rRNA	1900	6.5 × 10^5
16S rRNA (*E.coli*)	1500	5.1 × 10^5
5S rRNA (*E. coli*)	120	4.1 × 10^4
tRNA (*E. coli*)	75	2.5 × 10^4

Mass of Nucleic Acid ↔ Moles of Nucleic Acid

Mass	*Moles*
1 μg/ml of nucleic acid	3.0 μM phosphate
1 μg of a 1 kb DNA fragment	1.5 pmol; 3.0 pmol ends
0.66 μg of a 1 kb DNA fragment	1 pmol

Sizes of Various Genomes

Organism	*Approximate Size (million bases)*
Human	3000.0
M. Musculus (mouse)	3000.0
Drosophila (fruit fly)	135.6
Arabidopsis (plant)	100.0
C. elegans (round worm)	97.0
S. cerevisiae (yeast)	12.1
E. coli (bacteria)	4.7
H. influenzae (bacteria)	1.8

Genomic Equivalents of Species

Organism	*Source of DNA*	*pg/haploid*[a] *Genome*	*Avg.*[b]	*µg quantity for Genome Equivalence*	*Number of Genomes* × 10^6
Human	diploid	3.50	3.16	10.0	2.86
Mouse	diploid	3.00	3.21	8.57	2.86
Rat	diploid	3.00	3.68	8.57	2.86
Bovine	haploid	3.24	3.24	9.26	2.86
Annelid	haploid	1.45	1.45	4.14	2.86
Drosophila	diploid	0.17	0.18	0.486	2.86
Yeast	haploid	0.016	0.0245	0.0457	2.86

[a] pg/haploid genome was calculated as a function of the tissue source. Genomic equivalence was calculated given that 10 µg of human genomic DNA contains 2.86 × 10^6 genome copies.

[b] Average of all values given in each tissue for that species.

3 Appendix

Simple UNIX Commands

The following tables contain a brief list of simple but useful UNIX commands[1]. These commands can be used to move around the file system, examine files, and copy, delete, or rename files. They can also be used to do housekeeping on a user's account, and to communicate with other users on the local system or on remote systems.

Directory Operations

Command	*Action*
pwd	present working directory (show directory name)
cd	change directory: **cd /path/name**
cd	change to your home directory: **cd**
mkdir	make (create) new directory: **mkdir Name**
rmdir	remove directory (if empty): **rmdir Name**
quota	check disk space quota: **quota -v**

File Operations

Command	*Action*
ls	list files
cp	copy files: **cp /path/name newname**
rm	remove (i.e. delete) files: **rm name**
mv	move or rename files: **mv name newname**
more	page file contents (spacebar to continue): **more name**
cat	scroll file contents: **cat name**
less	better pager than more? (**q** to quit): **less name**
vi	visual text editor (**:wq** to save and quit): **vi name**
pico	pico text editor (**Ctrl-X** to quit): **pico name**
chmod	change mode of file permissions: **chmod xxx name**

[1] Most commands have options. To see what options are available, use the **man** command to open the manual pages for that command, e.g. type **man ls** to open the manual for the **ls** command.

Manual Pages

Command	*Action*
man	open the man pages for a command: man command

Communications

Command	*Action*
write	write messages to another user's screen
talk	talk split-screen with another user: **talk username**
mail	UNIX email command
pine	send or read E-mail with pine mail system
telnet	connect to another computer via the network
ftp	file transfer over the network
lynx	text-based Web browser

System Operations

Command	*Action*
df	show free disk space
du	show disk usage
ps	list your processes
kill	kill a process: **kill ###**
passwd	change your password
date	show date and time
w	who is doing what on the system
who	who is connected to the system
ping	ping another computer (is it alive?)
finger	get information on users
exit	exit, or logout, from the system

X Windows

Command	*Action*
clock &	display a clock (&: run in background)
cmdtool &	command tool window
filemgr &	file manager
mailtool &	email program
perfmeter &	system performance meter
seqlab &	SeqLab interface for GCG
setenv DISPLAY	for setting the DISPLAY environment variable
shelltool &	shell tool window
textedit &	text editor
xterm &	X terminal window

Index

Limited Warranty and Disclaimer

Humana Press Inc. warrants the CD-ROM contained herein to be free of defects in materials and workmanship for a period of thirty days from the date of the book's purchase. If within this thirty day period Humana Press receives written notification of defects in materials or workmanship, and such notification is determined by Humana Press to be valid, the defective disk will be replaced.

In no event shall Humana Press or the contributors to this CD-ROM be liable for any damages whatsoever arising from the use or inability to use the software or files contained therein.

The authors of this book have used their best efforts in preparing this material. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. Neither the authors nor the publisher make warranties of any kind, express or implied, with regard to these programs or the documentation contained within this book, including, without limitation, warranties of merchantability or fitness for a particular purpose. No liability is accepted in any event, for any damages including incidental or consequential damages, lost profits, costs of lost data or program material, or otherwise in connection with or arising out of the furnishing, performance, or use of the programs on this CD-ROM.

System Requirements for CD Software

BioDiscovery

The following hardware and software is required to successfully install and run BioDiscovery software:

- Operating System: Micrsoft Windows 9x, 2000, or NT4.
- Processor: An IBM-PC or equivalent with a Pentium 500 MHz or higher. (Pentium 800 or higher recommended.)
- Monitor: A SVGA or higher video system; 1024x768 or higher recommended.
- Random Access Memory (RAM): At least 256 MB of RAM; 512MB or more recommended.
  NOTE: Program performance may suffer without adequate RAM. Since virtual memory contained on the hard disk will not increase performance, it is highly recommended that your system have a sufficient amount of RAM installed.
- Hard Drive: At least 70 MB of free hard disk space; 150 MB recommended.
  NOTE: The amount of hard disk space listed above does not include the Java Runtime Environment, which may also need to be installed. If you do not have the Java 1.3 Runtime Environment, you will need an additional 10 MB for this installation.
- CD-ROM: A CD-ROM Drive is required to install the program.

ClustalX

Versions for both IBM-PC (Windows) and Macintosh are included. The files in the packages will need to be unpacked with common unzipping utilities.

Ensembl

Running the Ensembl web site is rather taxing on hardware. At a minimum you will need:

- A UNIX-like OS (e.g., Tru64, Solaris, Linux, etc).
- 8 Gigabytes of free hard disk to hold the full MySQL databases, and about the same again for downloading and unpacking the data.

As with most systems, the more memory and processor speed you have, the better. As an indication for those looking to purchase machines, our webservers are 3 Compaq DS20s, with 2Gb main memory and 2 Alpha EV6 500MHz processors.

MicroAnalyser

This software may be installed on Macintosh computers running Mac System 9 or Mac OS X.

Oligo

Demo versions of the Oligo primer design and analysis software will run on both IBM-PC (Windows) and Macintosh computers.

Sequencealign

This folder contains a PowerPoint demonstration of sequence alignment that will run on both IBM-PC (Windows) and Macintosh computers with Microsoft PowerPoint.

Singh_perl_scripts

This folder contains perl scripts for statistical analysis that can be run on any computer on which perl is installed.

Staden

The Staden Sequence Analysis Package and the Gap4 Viewer software can be installed on an IBM-PC computer running Microsoft Windows 98 or later.

TreeView

Versions of TreeView for both IBM-PC (Windows) and Macintosh computers are included.

Introduction to **Bioinformatics**

A Theoretical and Practical Approach

Includes CD-ROM

Edited by

Stephen A. Krawetz
Wayne State University School of Medicine, Detroit, MI

David D. Womble
Wayne State University School of Medicine, Detroit, MI

The genomic revolution that has spawned microarrays and high throughput technologies has produced vast amounts of complex biological data that require integration and multidimensional analysis. In *Introduction to Bioinformatics: A Theoretical and Practical Approach,* leading researchers and experts introduce the key biological, mathematical, statistical, and computer concepts and tools necessary for physical and life scientists to understand and analyze these data.

For physical and computer scientists, the book provides a sound biological framework for understanding the questions a life scientist would ask in the context of currently available computational tools. Here, the basic cellular structure and the biological decoding of the genome, the long-range regulation of the genome, the *in silico* detection of the elements that impact long-range control, and the molecular genetic basis of disease as a consequence of replication are explained. Reviews of clinical human genetics, the various clinical databases, and pertinent issues in population genetics complete this tutorial.

For life scientists, a complete discussion of the UNIX operating system, which is commonly used to support advanced computational tools, offers biologists graphical-user-interface comfort in a command-line environment, plus an understanding of the installation and management of UNIX-based software tools.

It is in the applications sections that the book provides a common meeting ground for life and physical scientists. Here they will find examples of the management and analysis of DNA sequencing projects, the modeling of DNA as a statistical series of patterns, various methods of pattern discovery, protein visualization, and the use of multiple sequence alignment to infer both functional and structural biological relationships. An accompanying CD contains several full and limited-trial versions of the programs discussed in the text, as well as a complete set of illustrations from each chapter specifically prepared for use as illustrative instructional material for lectures and presentations.

Cross-disciplinary and highly instructive, *Introduction to Bioinformatics: A Theoretical and Practical Approach* offers life, mathematical, computer, and statistical scientists an integrated foundation for using today's many new computational advances to elucidate the nuances of both genetic code and integrated biological systems.

Features

- Understandable introduction to bioinformatics for both life and physical scientists
- Introduction to computer and biostatistical principles suitable for life scientists
- Overview of relevant biological principles and systems suitable for the physical scientist
- Review of state-of-the-art bioinformatics tools and methods
- CD containing color graphics, lecture material, and software ready for use
- DNA sequence assembly and analysis using the Staden software suite
- Analysis of microarray data using biostatistical techniques
- Numerous examples of practical applications
- Resources for Whole Genome analysis

ISBN 1-58829-064-6

90000

9 781588 290649

Introduction to Bioinformatics
A Theoretical and Practical Approach
ISBN:1-58829-064-6 E-ISBN: 1-59259-335-6

humanapress.com