Renato Bruni
Editor

Mathematical Approaches to Polymer Sequence Analysis and Related Problems

Springer

Mathematical Approaches to Polymer Sequence Analysis and Related Problems

Renato Bruni
Editor

Mathematical Approaches to Polymer Sequence Analysis and Related Problems

Springer

Editor
Renato Bruni
Department of Computer and System Sciences
University of Roma “Sapienza”
Via Ariosto 25
00185 Roma
Italy
bruni@dis.uniroma1.it

ISBN 978-1-4419-6799-2 e-ISBN 978-1-4419-6800-5
DOI 10.1007/978-1-4419-6800-5
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010938717

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Many problems arising in biological, chemical, and medical research, which could not be solved in the past due to their dimension and complexity, are nowadays tackled by means of automatic elaboration. Powerful computers are indeed used intensively for solving many problems having biological origin, thus creating the emerging field of science called "bioinformatics." However, the success of such approaches depends not only on brute computational strength of those computers, but also, and often critically, on the mathematical quality of the models and of the algorithms underlying those solution procedures.

Solving a problem may be seen as converting information, in such a way that the solution of the problem (information in output) is extracted from its description (information in input), possibly passing through a number of intermediate states. By adopting this view, information handled when dealing with many of the above-mentioned problems becomes, at some stage, a sequence. Nature often encodes relevant information into sequences. Therefore, a central role in bioinformatics is played by sequence analysis problems or by the related problems of analyzing the effects or the behavior of some sequence.

The present volume offers a detailed overview of some of the most interesting mathematical approaches to sequence analysis and other sequence-related problems. Special emphasis is devoted to problems concerning the most relevant biopolymers (proteins and genetic sequences), but the exposition is not limited to them. A considerable effort has been made to render the volume comprehensible to researchers coming from either of the two hemispheres of bioinformatics: mathematics and computer science on one side, and biology, chemistry, and medicine on the other.

Rather than an exhaustive coverage of the topic, which would be clearly impossible to do in just one book, the volume is intended as a snapshot of the latest research development and of the potentialities that operations research and machine learning techniques bring in this interdisciplinary field of research. Moreover, the volume aims at bridging the two mentioned halves of bioinformatics that are still quite disjoint, promoting a cross-fertilization hopefully fostering future research in the field.

Primary selection criterion for the chapters has been scientific quality and importance. Additional selection criteria have been: (1) considering only approaches having a nontrivial mathematical basis; and (2) providing up to date contents not already largely available in other books published on similar subjects.

Organization of the Volume

Due to the wide heterogeneity of the matter, from the point of view of both problems considered and techniques presented, it may be useful to the reader tracing the following short sketch of the volume organization.

The first part of the volume deals with problems originating from the study of protein sequences. Proteins and peptides are polymers made from units called amino acids, and a basic problem is the determination of their amino acid sequence when that is unknown. This is sometimes called analysis of the primary structure. In Chap. 1, Bruni deals with this problem, with a focus on peptides, since proteins are essentially polypeptide chains, and describes exact and complete approaches based on propositional logic.

To be able to perform their biological functions, proteins fold into specific spatial conformations. Another relevant problem is the determination of such structures, known as the problem of protein structure analysis or prediction. In particular, the disposition of highly regular substructures in the protein sequence, such as helices, sheets, and strands, is called the secondary structure, while the three-dimensional structure of a single protein molecule, and the spatial arrangement of the above-mentioned elements of the secondary structure, is called the tertiary structure.

In Chap. 2, Di Lena et al. describe approaches to protein structure analysis based on decomposition, with specific attention to the secondary structure prediction and the protein contact map prediction by means of machine learning techniques. In Chap. 3, Patrizi et al. tackle again the problem of secondary structure prediction, performing a classification by means of nonlinear binary optimization techniques, with the aim of detecting isoform proteins considered as markers in oncology. Similarly, in Chap. 4, Biba et al. describe approaches to the protein folding prediction by modeling the sequence by means of Markov logic networks, that is, networks obtained by introducing probability in first-order logic.

The volume then gradually moves to problems originating from the study of genetic sequences. Deoxyribonucleic acid, or DNA, is a long polymer made from repeating units called nucleotides. It contains the genetic instructions used in the development and functioning of all known living organisms. In Chap. 5, Ceci et al. deal with the problem of discovering motifs, that are sequence patterns frequently appearing in DNA, RNA, or proteins, and therefore probably having specific biological functions. They are discovered by mining association rules in the three-dimensional space.

In Chap. 6, Mosca and Milanesi consider the problem of studying intermolecular interactions among DNA, RNA, and proteins obtained by means of sequence analysis techniques. When viewing those interactions at a system level, the dynamics of biochemical pathways can be simulated, and therefore better understood, by means of mathematical models.

In Chap. 7, Graça et al. deal with the problem of determining haplotype information, that is, genetic information inherited from ancestors, from genotype information, that is, all the genetic constitution of an individual, using approaches based on propositional logic. On related themes, in Chap. 8, Catanzaro describes the problem of calculating phylogenies, that is, graphs representing the evolutionary relationships among species. Several optimization models for estimating them from molecular data such as DNA and RNA under different paradigms are explained and discussed.

In Chap. 9, Salvi et al. tackle the problem of performing studies of human genome by means of data mining techniques, known as genome-wide association studies, for a stratified population. This means that the individuals of the population are not uniform but carry different genetic backgrounds, and this often produces false association results. The effects of different statistical techniques are considered to devise an efficient strategy for overcoming this problem.

The last part of the volume considers problems originating from the study of polymers not having biological origin. Polymerization reactions can be divided into: (i) addition polymerization, producing the so-called addition polymers (also classified as chain-growth polymers, with some exceptions), which grow one monomer at a time, and (ii) condensation polymerization, producing the so-called condensation polymers (also classified as step-growth polymers), which grow eliminating small molecules during the synthesis. In Chap. 10, Montaudo deals with the problem of predicting the sequence distribution of addition polymers; while in Chap. 11, Montaudo discusses the same problem for condensation polymers, using in both a variety of mathematical techniques.

Rome, Italy
March 2010

Renato Bruni

Contents

Chapter 1
Complete and Exact Peptide Sequence Analysis Based on Propositional Logic

Renato Bruni

Abstract Peptides are the short polymeric molecules constituting all the proteins. They are formed by the linking of amino acids, and the determination of the amino acid sequence of a peptide is a fundamental issue in many areas of chemistry, medicine and biology. Nowadays, the prevalent approach to this problem consists in using a mass spectrometry analysis. This gives information about the molecular weight of the full peptidic molecule and of its fragments. Such information should be used in order to find the sequence, but this constitutes, in the general case, a difficult mathematical problem. After a brief overview of the approaches proposed in literature, and of their features and limits, the chapter describes in detail a promising one based on propositional logic. Differently from the others, this approach can be proved to be complete and exact.

1.1 Introduction

Peptides are short polymeric molecules formed by the linking of components called *amino acids* by means of covalent bonds called *peptide bonds*, in order to form a *chain*. Proteins are polypeptide chains; they are formed by a similar linking of amino acids, but the chain is generally longer. There are several different conventions to determine this distinction, see e.g. [4, 32].

The determination of the *sequence* of amino acids forming a peptide or a protein is one of the most important and frequent issues in many areas of chemistry, medicine and biology, as well as in several other applicative fields. In the case of peptides, this is often called de novo sequencing, whereas in the case of protein, this is often called determination of the primary structure. However, proteins are generally too extended for performing an accurate sequence analysis on the whole chain in a single step. Therefore, a protein molecule is usually divided into

R. Bruni (✉)
Department of Computer and System Sciences, University of Roma "Sapienza", Italy
e-mail: bruni@dis.uniroma1.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_1,

its component peptides (via enzymatic digestion and subsequent fractionation with HPLC or capillary electrophoresis, [32]), and the original analysis is converted into a number of peptide analyses which are performed individually. It is worth noting that this problem has a theoretical structure that is able to represent various other problems of sequence analysis. At the very basic level, there is a set of possible components that are individually known a chain formed by some of such components, possibly repeated, whose sequence is not known and that cannot be inspected directly; and the aim is to determine this sequence of components forming the chain.

Nowadays, a widely used and well-established approach to peptide sequence analysis consists in the use of mass spectrometry [19, 20, 23, 28]. Such technique can provide the absolute molecular weight distribution of a number of molecules in the form of a *spectrum*: for each molecular weight, the amount of material having that molecular weight produces a *peak* having a certain *intensity*. The study of the weight pattern in the spectrum can be used for understanding the structure of such molecules, especially when using the mass spectrometry/mass spectrometry methodology (also known as MS/MS, or tandem mass, [29]). This procedure works as follows. After the first mass analysis, some molecules of the protonated peptide under analysis, called *precursor ion*, are selected and collided with other non-reactive elements. This interaction leads to the fragmentation of many of such molecules, and the collision-generated decomposition products undergo a second mass analysis. Therefore, such analysis provides the absolute molecular weight of the full precursor ion, as well as those of the various ionized fragments obtained from that precursor ion. Non-ionized fragments, on the contrary, do not appear in the spectrum. Such experiments may be performed using several instrumental configurations, mainly triple quadrupole (QQQ), quadrupole time-of-flight (Q-TOF) and ion trap devices [20].

Since the weights of the possible components are known, and rules for determining the weights of sequences of known composition are available, the MS/MS information could be used in order to determine the unknown sequence of a peptide. This is, however, a difficult mathematical problem, as explained in detail in Sect. 1.2. Note that the presence of fragments constitutes the only source of information about the inner structure of the molecule under analysis: in the absence of fragmentation, the inner structure would be unknown. Several approaches to this problem have been proposed, as reported in Sect. 1.3. In particular, a promising approach [5] is based on a propositional logic modeling [12, 18, 31] of the problem, as explained in Sects. 1.4 and 1.5. It can be shown that all and only the possible outcomes of a sequence analysis can be obtained by finding all models of a propositional logic formula. The off-line computation of the so-called weights database, which substantially speeds-up the sequencing operations, is described in Sect. 1.6. This is obtained by finding a correspondence between sequences and natural numbers, so that all sequences up to a certain molecular weight can be implicitly considered in the above database, and explicitly computed only when needed. The procedure is illustrated by considering the case of peptides, but may be adapted to generic polymeric compounds submitted to mass spectrometry. Results on real-world problems, shown in Sect. 1.7, demonstrate the effectiveness of this approach.

1.2 From the Spectrum to the Sequence

The MS/MS spectrum contains our information about the structure but does not have any direct reference to the components of the polymer, being a mere succession of peaks corresponding to different molecular weights. The intensity of each peak is proportional to the number of molecules having that weight in the sample under analysis. A typical example is observable in Fig. 1.1. Further processing is then requested.

An initial *peak selection* phase is needed. This is generally done by removing all peaks below a certain intensity, since they are too noise-prone to be considered significant, and by considering informative all other peaks. After this phase, the higher molecular weight among informative peaks is the one of the full polymer under analysis, whereas the others correspond to its fragments. Though fragmentation is a stochastic process, some rules may be traced. The most abundant fragments are generally given by the cleavage of the weakest molecular bonds. Therefore, some types of fragments, called *standard* fragments, are more common than others and should more likely correspond to the peaks selected as informative in the spectrum. In the case of peptides, for instance, there are six different types of standard fragments, called a, b, c, x, y and z. Fragments appear in the spectrum when ionized by retaining one or more electrical charges. Unfortunately, when analyzing each of such fragment peaks, we neither know the type of fragment that originated it (it could be either any of the standard types or also a non-standard type) nor the number of electric charges that this fragment retained.

Now, some analysis techniques search for specific weight patterns in the spectrum and check them against similar patterns available from a databases of compounds [17]. However, when our compound is not in the databases (which may very well happen) or when the it differs from the standard known form (protein sequences, for instance, often undergo modifications), a constructive identification is required. Constructive identification, however, is not immediate, and, moreover, the information contained in the spectrum may be insufficient for a univocal identification.

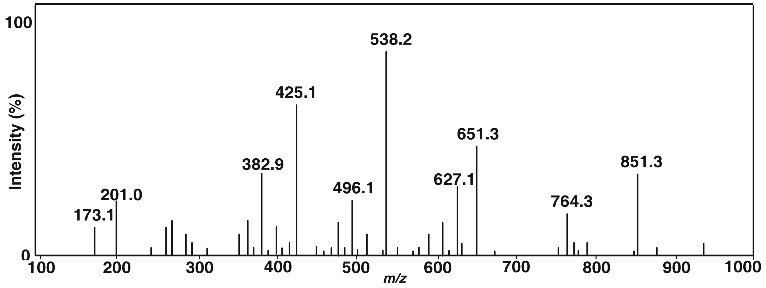


Fig. 1.1 A MS/MS spectrum generated by collision-induced dissociation

Definition 1.1. We will say that a sequence of components is *compatible* with a given spectrum if every informative peak in the spectrum admits an interpretation as a standard fragment of that sequence.

Often, however, there exists more than one sequence which is perfectly compatible with a given spectrum. This means that the spectrum does not contain enough information to determine uniquely the sequence, and so there are more possibilities. Consider, for instance, the case of an incomplete fragmentation: if a part of a polymer never did break in the analysis, no detailed information on the inner structure of that part can be achieved. In this case, all the possible sequences compatible with the spectrum should be found, so as to guarantee accurate and objective character of the analysis. Sometimes it may also happen that a spectrum contains one or more peaks that have been selected as informative, but are instead due for instance to noise, non-standard fragmentation or spurious components. They are therefore not interpretable as standard fragments; hence, it may be the case that not even a sequence exists which is compatible with the given spectrum. In this case, the best that can be done, informally speaking, is being compatible with as many peaks as it is possible.

Definition 1.2. A sequence of components is *ν-compatible* with a given spectrum if every informative peak in the spectrum, except a number ν of them, admits an interpretation as a standard fragment of that sequence. This number of uninterpreted peaks will be called the *mismatch* number ν.

In order to analyze the features of the various approaches to the problem of passing from the spectrum to the sequence, we need to define the following sets.

Definition 1.3. The *resolvents* of a spectrum are all the sequences that are compatible with the that spectrum (but are not given: are those that should be found).

Definition 1.4. The *results* of a procedure are all the sequences that are given as the outcome of the analysis procedure.

The above two sets may coincide or not, depending on the quality of the adopted solution approach.

Definition 1.5. A solution approach is said to be *complete* if it guarantees finding as results all the possible resolvents of the spectrum; *incomplete* when such guarantee cannot be given, and therefore a part of the possible resolvents may be neglected. This could mean finding, in some cases, no resolvents at all.

Definition 1.6. A solution approach is said to be *exact* if it guarantees that every result given by the analysis is perfectly compatible with the given spectrum; *approximate* when this cannot be guaranteed, and therefore the results given are only near-compatible, according to some nearness criterion.

A result given by an approximate procedure may just leave some informative peaks without an interpretation as standard fragments, or may give interpretation that are not numerically precise. Note that this concept of approximate results is more general and less precise than that of ν-compatible solution. Nevertheless, due to the stochastic aspects involved in the fragmentation process, these approximate results may sometimes be probable solutions.

Completeness and exactness are clearly positive features for a solution approach. However, complete and exact methods generally require larger computational times than incomplete or approximate ones [17, 21]. Note also that a complete and exact procedure correctly produces no results (or only results with mismatch $\nu > 0$) when the spectrum has no resolvents.

1.3 Different Approaches to the Problem

For that which concerns constructive peptide sequencing, known as de novo sequencing, some analysis procedures have been developed and implemented in a number of software systems, e.g., DeNovoX [24], Mass Seq [25], Peaks [26] and Spectrum Mill [27]. Each of such procedures is essentially based on one of the following two approaches.

The first one consists in searching the spectrum for continuous series of fragments belonging to the same standard type and differing by just one amino acid, which is therefore identified. The whole sequence can be obtained in this manner when the spectrum contains a complete series of fragments. This, however, is often unlikely to occur. Since the fragmentation process is a stochastic one, though peptides tend to break at the conjunction of amino acids, they usually do not break at every conjunction of amino acids, and furthermore such cleavages may be of any of the different types mentioned. And, if the collision energy is increased, the peptide produces more fragments, but may also break at locations that are not the conjunction of amino acids, producing some non-standard fragments. Hence, every result given by the procedure is guaranteed to be a resolvent of the spectrum. On the contrary, there could be many resolvents of the spectrum not obtained as results because of the incompleteness of the series of fragments. The above approach should therefore be classified as heavily incomplete, though exact.

The second approach consists in iteratively generating, using Monte Carlo methods [8], a large number of virtual sequences and evaluating the match of the corresponding (theoretical) mass patterns with the (actual) mass pattern of the spectrum under investigation. Therefore, sequences producing a spectrum similar to the one under analysis can be obtained, but no completeness can be guaranteed. The number of possible peptides is in fact very large: just for example, the possible peptides composed of 12 amino acids, choosing them among 20 possible amino acid types, are $20^{12} \approx 10^{15}$. Hence, even hypothesizing of generating and checking 10^5 sequences per second, which for nowadays computer seems quite optimistic, after 10^4 seconds of computation (almost 3 hours), only 10^9 sequences would have been tried, which means a relatively small part of the possible ones (one every 10^6 in the example). Therefore, only a negligible portion of the solution space would have been explored, and there could be many sequences producing a spectrum much more similar to the one under analysis that have not been considered. And, even by protracting the search or increasing the search speed, when the number of generated sequences becomes near to the number of possible ones, no guarantee of repeating the same sequences can be given. This would require either memorizing all the tested ones and checking all of them after the generation of each new one, which is clearly impossible to do in reasonable times for nowadays computer technology [15], or generating them in some ordered manner, and not by means of Monte Carlo methods. Finally, the similarity of spectra must be evaluated, by choosing some similarity criterion, with the consequence that the approach becomes an approximate one. The above described analysis techniques suffer therefore from considerable structural limitations.

Due to its combinatorial nature, the problem has also been recently approached by means of discrete mathematics. Specifically for the peptide sequencing problem, there have been, on one hand, the graph theoretical construction proposed in [14], which evolved into the dynamic programming algorithms proposed in [2, 11], and, on the other hand, the branching-based algorithm proposed in [7], which evolved into the propositional logic modeling proposed in [5]. The first approach has the advantage of requiring a computational time for finding each solution which is polynomial, hence tractable [15], when imposing some limitations to the problem, namely no multi-charged fragments can appear in the spectrum, and only peaks corresponding to a set of fragment types which is "simple" [2] (e.g., only a-ions, b-ions and y-ions) can appear in the spectrum. When overriding such limitations, polynomial time cannot be guaranteed, and in any case the procedure cannot work with a spectrum in which all types of fragments and charges may appear. The problem in the general case is, however, NP-complete [2]. The second approach, on the other hand, has no structural limitations regarding types of fragments and charges, and performs a complete search. It requires, however, a heavier computational load; but can be improved as described in the rest of the chapter.

1.4 A Mathematical View of the Fragmentation Process

When a polymer undergoes a MS/MS analysis, the occurring fragmentation process gives an essential support to the sequencing. We now analyze in detail peptide fragmentation. Similar analyses may be performed of course also for other categories of polymers. Peptides basically are single sequences of building-blocks called *amino acids*. Each amino acid molecule has the following general chemical structure.

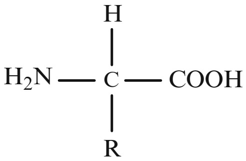


Table 1.1 Commonly considered amino acids

Name	Abbreviations		Molecular weight	Limitations
Glycine	Gly	(or G)	75.07	–
Alanine	Ala	(or A)	89.34	–
Serine	Ser	(or S)	105.10	–
Proline	Pro	(or P)	115.14	–
Valine	Val	(or V)	117.15	–
Threonine	Thr	(or T)	119.12	–
Cysteine	Cys	(or C)	121.16	–
Taurine	Tau		125.15	Only C-terminal
Piroglutamic acid	pGlu		129.10	Only N-terminal
Leucine	Leu	(or L)	131.18	–
Asparagine	Asn	(or N)	132.12	–
Aspartic acid	Asp	(or D)	133.11	–
Glutamine	Gln	(or Q)	146.15	–
Lysine	Lys	(or K)	146.19	–
Glutamic acid	Glu	(or E)	147.13	–
Methionine	Met	(or M)	149.22	–
Histidine	His	(or H)	155.16	–
Phenylalanine	Phe	(or F)	165.16	–
Arginine	Arg	(or R)	174.21	–
Tyrosine	Tyr	(or Y)	181.19	–

There is a large number of possible amino acids, differing in the internal chemical structure of the radical R, and, therefore, for their functional characteristics and their molecular weights. Many of them cannot be specified in the genetic code; hence, the most commonly considered ones generally include the 20 reported in Table 1.1. Moreover, each amino acid may present one of the many possible modifications, such as phosphorylation, acetylation and methylation. This would produce alterations to its standard molecular weight. Note also that the equivalent mass involved in the molecular bindings leads to non-integer values for the amino acid weights and that the very weight of each amino acid type is not a single fixed value, but may assume different values depending on the presence of different isotopes of the various atoms constituting the amino acid. Values reported in Table 1.1 are just the average masses of the molecules.

An accurate and general sequencing procedure should be able to deal with the above uncertainties, by taking as part of the problem data the information about:

- Which are the components that should be considered as possible for the current analysis;
- Their weight values (in *unified atomic mass units* u, or daltons);
- Possible limitations on the position they can assume within a peptide chain;
- The desired numerical precision of the sequencing procedure, set on the basis of the accuracy of the adopted mass spectrometry device;
- And any other incidentally known information.

When performing a sequence analysis, the solution is obviously not known in advance. However, we often know which aspects of the solution can be considered as possible for the current analysis, and which ones cannot. For instance, we may know that a peptide under analysis contains at least a certain number of molecules of some amino acid or does not contain another amino acid, etc. At worst, if nothing else is known, simply every generic aspect of the solution should be considered as possible.

This may be formalized by evaluating the number n of possible components (the amino acids) that must be considered for the current analysis, the set $N = \{1, 2, \dots, n\}$ of the indices i corresponding to such components in increasing weight order, the set

$$A = \{a_1, a_2, \dots, a_n\}, \qquad a_i \in \mathbb{R}_+$$

of the weight values of such components (the molecular weights of the amino acids) that must be considered for the current analysis, together with the sets

$$\begin{aligned} \text{Min} &= \{m_1, m_2, \dots, m_n\},\ m_i \in \mathbb{Z}_+ \\ \text{Max} &= \{M_1, M_2, \dots, M_n\},\ M_i \ge m_i,\ M_i \in \mathbb{Z}_+, \end{aligned}$$

respectively, of the minimum and the maximum of the possible number of molecules of each component that must be considered for the current analysis, the number d of decimal digits that can be considered significant for the current analysis, and a value $\delta \in \mathbb{R}_+$ of the maximum numerical error that may occur in the current analysis.

Amino acids can link to each other into a peptidic chain by connecting the aminic group NH_2 of one molecule with the carboxyl group COOH of another molecule. The free NH_2 extremity of the peptide is called N terminus, while the free COOH extremity is called C terminus. Some amino acids, especially the modified ones, can be situated only in particular positions of the sequence, i.e., only N-terminal or only C-terminal. Since each of the peptidic bonds releases an H_2O molecule, the weight of a peptide is not simply the sum of the weights of its component amino acids. Moreover, the weights observed in the spectrum correspond to the actual weights only for the ionized molecules (ions) which retain one single electrical charge. When, on the other hand, an ion retains more than one charge, the weight observed in the spectrum is only a fraction of the actual ion weight. By considering the set

$$Y^0 = \{y_1^0, y_2^0, \dots, y_n^0\}, \qquad y_i^0 \in \mathbb{Z}_+$$

of the numbers of molecules of each component (here the amino acids) contained in the overall polymer (here the peptide), and the number $e_0 \ge 1$ of electrical charges retained by the ionized peptide, the observed weight w_0 of the overall peptide is given by the following equation:

$$w_0 = \frac{\sum_{i \in N} (y_i^0 (a_i - c_a)) + c_a + c_0 e_0}{e_0} \pm \delta, \tag{1.1}$$

where c_a and c_0 are constant values. When considering $d = 3$ decimal digits, c_a is 18.015 and c_0 is 1.008.

Example 1.1. A small peptide with sequence Leu-His-Cys-Thr-Val ionized by only one charge, considering only $d = 2$ decimal digits, has an observed weight of $w_0 = (131.18 - 18.02) + (155.16 - 18.02) + (121.16 - 18.02) + (119.12 - 18.02) + (117.15 - 18.02) + 19.02 \pm \delta = 572.69 \pm \delta$.

Several different types of fragments can be obtained during the fragmentation process. In particular, there are three possible standard N-terminal ionized fragments, called a-ion, b-ion and c-ion, and three possible standard C-terminal ones, called x-ion, y-ion and z-ion, as illustrated in Fig. 1.2. Note that b-ions and y-ions are generally the most common.

Again, each fragment has a weight which is not simply the sum of those of its component amino acids. By considering the number f of fragment peaks selected in the spectrum; the set $F = \{1, 2, \ldots, f\}$ of the indices j corresponding to such peaks in decreasing weight order; the set

$$W = \{w_1, w_2, \ldots, w_f\}, \qquad w_j \in \mathbb{R}_+$$

of the weights corresponding to such peaks (so that w_0 remains the weight of the overall peptide); the sets

$$Y^j = \{y_1^j, y_2^j, \ldots, y_n^j\}, \qquad y_i^j \in \mathbb{Z}_+ \qquad j = 1, \ldots, f$$

of the numbers of molecules of each component contained in the fragment of weight w_j, $j = 1, \ldots, f$; the number $t_{\max}$ of all the possible standard types of fragments that should be considered for the current analysis; the set

$$T = \{1, 2, \ldots, t_{\max}\}$$

of the indices t corresponding to such types; the maximum number of electrical charges $e_{\max}$ that a ion may retain in the current analysis; the set

$$E = \{1, 2, \ldots, e_{\max}\}$$

of the numbers e of electrical charges that a ion may retain in the current analysis; the type $t_j \in T$ of the fragment of weight w_j, $j = 1, \ldots, f$; the number $e_j \in E$ of electrical charges retained by the fragment of weight w_j, $j = 1, \ldots, f$, the relation that can be observed in the spectrum between the weight of each fragment and the weights of its components is the following.

$$w_j = \frac{\sum_{i \in N} [y_i^j (a_i - c_a)] + c_t + c_0 e_j}{e_j} \pm \delta, \qquad j = 1, \ldots, f \tag{1.2}$$

Values c_a and c_0 are as above, and c_t is a constant value depending on the type t_j of the fragment. When considering $d = 3$ decimal digits, c_t is -28.002 for a-ions, 0.000 for b-ions, 17.031 for c-ions, 44.009 for x-ions, 18.015 for y-ions and 1.992 for z-ions.

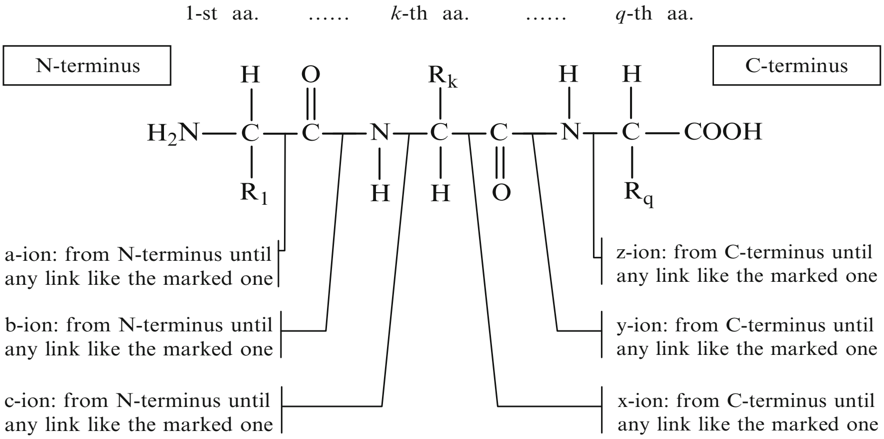

1-st aa.
......
k-th aa.
......
q-th aa.
N-terminus
C-terminus
H O R_k H H
H₂N—C—C—N—C—C—N—C—COOH
R_1 H H O R_q
a-ion: from N-terminus until any link like the marked one
b-ion: from N-terminus until any link like the marked one
c-ion: from N-terminus until any link like the marked one
z-ion: from C-terminus until any link like the marked one
y-ion: from C-terminus until any link like the marked one
x-ion: from C-terminus until any link like the marked one

Fig. 1.2 Different types of fragments obtainable from a peptide

In other words, the rules giving the weights of the six standard fragments having only one charge are as follows:

- a-ion weights the sum of its component amino acids, each of which decreased by 18.015, plus $(-28.002 + 1.008) = -26.994$;
- b-ion weights the sum of its component amino acids, each of which decreased by 18.015, plus $(0.000 + 1.008) = 1.008$;
- c-ion weights the sum of its component amino acids, each of which decreased by 18.015, plus $(17.031 + 1.008) = 18.039$;
- x-ion weights the sum of its component amino acids, each of which decreased by 18.015, plus $(44.009 + 1.008) = 45.017$;
- y-ion weights the sum of its component amino acids, each of which decreased by 18.015, plus $(18.015 + 1.008) = 19.023$;
- z-ion, finally, weights the sum of its component amino acids, each of which decreased by 18.015, plus $(1.992 + 1.008) = 3.000$.

Besides, additional (non-standard) fragmentation may also occur: losses of small neutral molecules such as water, ammonia, carbon dioxide, carbon monoxide, or breaking of a side chain. In such cases, the weight of the fragment decreases accordingly. Finally, since fragments appear in the spectrum only when they are ionized, the fact that a fragment is observed does not mean that its complement fragment will be observed as well.

Example 1.2. When considering the spectrum reported in Fig. 1.1 and making the simplifying hypothesis of selecting only the peaks labelled with numbers (even if in practice a slightly larger set of peaks should be considered), we have $w_0 = 851.3$, $f = 9$, and $W = \{764.3, 651.3, 627.1, 538.2, 496.1, 425.1, 382.9, 201.0, 173.1\}$.

1.5 A Logic Encoding of the Peak Interpretation Problem

Each peak of weight w_j selected from the spectrum must be of one of the types $t \in T$ and have a charge $e_j \in E$, but the exact type and charge is in general unknown. In other words, each peak may have several different *interpretations*. If a peak of weight w_j is considered for instance an a-ion, it may be originated by a certain amino acid sequence having a certain weight; if it is considered a b-ion, it cannot be originated by that sequence, but by another sequence having a different weight, and so on. Moreover, since there are rules about incompatibility of fragments and electrical charges of ions, not all of the interpretations are admissible: when interpreting one peak, the interpretations given to all other peaks must be considered. The peak interpretation problem is therefore a decision problem that should be solved by considering all peaks at the same time.

Definition 1.7. The *peak interpretation problem* consists of assigning to each peak w_j selected from the spectrum, $j = 1, \ldots, f$, (at least) one hypothesis about the type $t_j \in T$ and the charge $e_j \in E$ of the fragment that originated w_j in such a way that all interpretations given to all peaks are *coherent*.

Definition 1.8. A set of interpretations for a set of peaks is coherent when all those interpretations respect a number of logical *rules* formalizing our knowledge of the problem.

Rules holding for every analysis are the incompatibility and multicharge rules, which are given below. Other analysis-specific rules may be generated, as observed below. Note that each peak should have *at least* one interpretation, but not necessarily *only* one. A peak may in fact be originated by more than one type of fragment incidentally having the same observed weight, even if this happens very rarely in practice.

We formalize the peak interpretation problem by means of propositional logic. By denoting with $w_j \rightarrow t, e$ the fact that peak w_j is interpreted as being due to a fragment of type $t \in T$ and having an electrical charge $e \in E$, we consider for each interpretation of w_j a propositional variable

$$x_{j \rightarrow t,e} \in \{True, False\}, \qquad j \in F,\ t \in T,\ e \in E$$

When considering for instance the above six standard types of fragments obtainable from a peptide and a maximum electrical charge $e_{\max} = 2$, we have $T = \{1, 2, 3, 4, 5, 6\}$ and $E = \{1, 2\}$. The possible interpretations of a peak w_j are therefore 12, and this may be represented by means of the following clause containing 12 variables, which means: `peak` w_j `is of type 1 and has charge 1 or it is of type 2 and has charge 1 or ... or it is of type 6 and has charge 2.`

$$(x_{j \rightarrow 1,1} \vee x_{j \rightarrow 2,1} \vee \cdots \vee x_{j \rightarrow 6,1} \vee x_{j \rightarrow 1,2} \vee x_{j \rightarrow 2,2} \vee \cdots \vee x_{j \rightarrow 6,2})$$

Those clauses are called interpretation clauses. In order to get rid of the fact that the weight of peptides and of their fragments is not simply the sum of those of their component amino acids, we define now a different (theoretical) model of polymeric compound, as follows.

Definition 1.9. Given a (real) single charge peptide of observed weight w_0, the *normalized peptide* associated with it is a (theoretical) polymeric compound having weight $w_0 - (c_a + c_0)$. The possible components of such normalized peptide are (theoretical) components having the following weights (which are those that amino acids assume in the internal part of the peptidic chain)

$$\bar{A} = \{(a_1 - c_a), (a_2 - c_a), \dots, (a_n - c_a)\}$$

As a result, the weight of the normalized peptide, as well as the weights of its fragments, is simply the sum of those of its components. By the above definition, the normalization of a single charge real peptide of observed weight w_0 is composed by a number of molecules of each of the components in $\bar{A}$ equal to the number of molecules $Y^0 = \{y_1^0, y_2^0, \dots, y_n^0\}$ of each amino acid contained in the real peptide of observed weight w_0.

Example 1.3. The normalized peptide corresponding to the real peptide of weight 572.69 of Example 1.1 has a weight of (572.69 – 19.02) = 553.67, and its component have the following weights: (131.18 – 18.02) = 113.16, (155.16 – 18.02) = 137.14, (121.16–18.02) = 103.14, (119.12–18.02) = 101.10, (117.15–18.02) = 99.13. If such normalized peptide breaks for instance in Leu-His and Cys-Thr-Val, such fragments, respectively, have weights: (113.16 + 137.14) = 250.30 and (103.14 + 101.10 + 99.13) = 303.37.

We will consider for such normalized peptide the above described topological concepts of N-terminus, C-terminus, peptidic bonds, etc., in their intuitive sense, as if it was a real peptide.

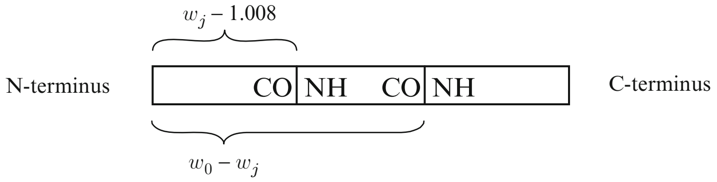
When a peak receives an interpretation, an hypothesis has been done about where the cleavage occurred in the peptide, and also about which was the chemical structure of the peptide in that point. Asserting that, for a single charge peptide of observed weight w_0, peak w_j is, for instance, a single charge b-ion means that starting from the N-terminus of the normalization of that peptide, there has been a cleavage between a CO and an NH, and that the part of such normalization going from the N-terminus to that cleavage has weight

$$w_j - 1.008 \pm \delta$$

On the contrary, asserting that, for the same peptide, the same peak w_j is now, for instance, a single charge y-ion means that starting from the C-terminus of the normalization of that peptide, there have been a cleavage between an NH and a CO and that the part of such normalization going from the C-terminus to that cleavage has weight $w_j - 19.023 \pm \delta$. Therefore, the part of the same normalization going from the N-terminus to that cleavage weights

$$w_0 - (c_a + c_0) - (w_j - 19.023) \pm \delta = w_0 - w_j \pm \delta$$

The two interpretations therefore bring to radically different hypothesis on the structure of the normalized peptide, as illustrated by the following diagram for $w_0 - (c_a + c_0) \approx 850$ and $w_j \approx 300$.

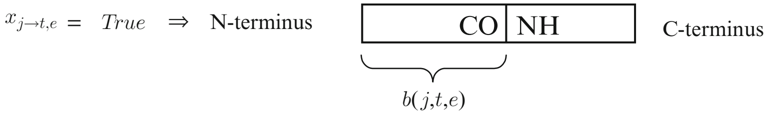


We now consider, for the each variable $x_{j\to t,e}$, with $j \in F$, $t \in T$, $e \in E$, the weight that the part of the normalized peptide going from the N terminus to the cleavage corresponding to interpretation $w_j \to t, e$ would assume.

Definition 1.10. An *N-terminal portion* of a normalized peptide is any part of that compound going from the N-terminus to any peptidic bond between CO and NH (a part that, if such bond was broken, would constitute a b-ion). The *hypothesized weight* of such N-terminal portion is the one given by the following function $b(j, t, e)$

$$b(j,t,e) = \begin{cases} (w_j - c_t - c_0 e_j)e_j & \text{for a-ions, b-ions, c-ions} \\ (w_0 - c_a - c_0 e_0)e_0 - (w_j - c_t - c_0 e_j)e_j & \text{for x-ions, y-ions, z-ions} \end{cases}$$

Note that charge e_0 of the precursor ion is known and fixed during each single analysis. By using the above concepts, variable $x_{j\to t,e} = \mathit{True}$ implies that there exists an N-terminal part of the normalized peptide having weight $b(j, t, e) \pm \delta$.



We are now able to introduce, in form of clauses, the additional sets of rules that an interpretation should respect in order to be coherent. A first one is the set of *incompatibility* rules. To this aim, we denote here variables using their corresponding values for b. Two variables $x_{b'}$ and $x_{b''}$ are incompatible if, for example, the difference between b' and b'' is smaller than the smallest possible component, that is:

$$|b' - b''| < (a_1 - c_a) - 2\delta$$

More generally, $x_{b'}$ and $x_{b''}$ are incompatible if the difference between b' and b'' has a weight value which cannot be any combination of possible components. In other words, it does not exist any non-negative integer vector $(y_1, y_2, \dots, y_n)^{tr} \in \mathbb{Z}^n_+$ verifying the following equation.

$$|b' - b''| = y_1(a_1 - c_a) + y_2(a_2 - c_a) + \cdots + y_n(a_n - c_a) \pm 2\delta$$

Therefore, incompatibility clauses of the following form are added for all the couples of incompatible variables $x_{b'}$ and $x_{b''}$.

$$(\neg x_{b'} \vee \neg x_{b''})$$

Another set of rules that should be considered in order to have a coherent interpretation is that of *multicharge* rules. Depending on the mass spectrometry device, ions retaining more than one electrical charge, called multicharged ions, are usually less common than single charged ions, and it is common practice to assume that if a multicharged ion has been observed in the spectrum, also the corresponding single charged one should appear in the spectrum. Therefore, each variable $x_{j' \to t,e}$ with $e > 1$ implies, if it exists, another variable $x_{j'' \to t,1}$ with $(j' - c_0 e)e = j'' - c_0$, as follows:

$$(\neg x_{j' \to t,e} \vee x_{j'' \to t,1})$$

Finally, a number of additional clauses representing a priori known information about the specific mass spectrometry device used for the analysis, about the analyzed compound or about other possibly known relations among the interpretations of the various peaks may also be generated. This is because, clearly, the more information can be introduced by means of clauses, the more reliable the results of the analysis will be.

By assuming no limitations on the structure of the generated clauses, therefore allowing the full expressive power of propositional logic, we obtain at this point a set of v clauses $C_1, C_2, \dots, C_v$. Generally, incompatibility clauses are by far the more numerous. Since all clauses must be considered together, we construct their conjunction, that is a generic propositional formula $\mathscr{F}$ in *conjunctive normal form* (CNF)

$$\mathscr{F} = C_1 \wedge C_2 \wedge \cdots \wedge C_v$$

Each truth assignment $\{$*True,False*$\}$ for the variables $x_{j \to t,e}$, with $j \in F$, $t \in T$, $e \in E$, such that $\mathscr{F}$ evaluates to *True* is known as a *model* of $\mathscr{F}$. We now have the following result.

Theorem 1.1. *Each model μ of the generated propositional formula $\mathscr{F}$ corresponds to a coherent solution of the peak interpretation problem for the peptide under analysis. Moreover, no coherent solution of the peak interpretation problem which does not corresponds to a model μ of $\mathscr{F}$ can exist.*

Proof. The proof relies on the fact that the formula $\mathscr{F}$ contains by construction all the rules (peak assignment rules, incompatibility rules, multicharge rules) that a peak's interpretation must satisfy to be considered coherent. Therefore, each model

μ gives an interpretation satisfying all the rules. Conversely, each interpretation satisfying all the rules corresponds to a truth assignment for the variables $x_{j \to t,e}$ such that $\mathscr{F}$ is *True*. □

Finding a model of a generic CNF, or proving that such model does not exist, is known as the *satisfiability* problem (SAT). Extensive references can be found in [10, 16, 18, 30]. This problem is NP-complete [15] in the general case. However, for the average size of generated instances, solution times of a DPLL branching algorithm are very moderate. Note also that in some special cases of peptide analysis, one may be able to obtain polynomially solvable formulae by imposing syntactical limitations on the structure of the generated clauses [3, 9, 13, 22]. For instance, when considering only b-ion and y-ion as the possible types of fragments, and only single charged ions, we obtain Quadratic formulae [1], which are polynomially solvable.

Since we are interested in all possible solutions of the peptide analysis, we are interested in all the possible coherent peaks interpretations, that is we are interested in finding all the models

$$\{\mu_1, \mu_2, \ldots, \mu_r\}$$

of $\mathscr{F}$. This was obtained in practice by modifying the SAT solver BrChaff [6] in such a way that, after finding a model, the search does not stop, but keeps exploring the branching tree until its complete examination.

Example 1.4. When considering the compound of Example 1.2 ($w_0 = 851.3$, $f = 9$, and $W = \{764.3, 651.3, 627.1, 538.2, 496.1, 425.1, 382.9, 201.0, 173.1\}$), the possible components of Table 1.1, and allowing a-ion, b-ion, c-ion, x-ion, y-ion, z-ion, and double and single charges, we obtain a formula $\mathscr{F}$ with 108 variables and 4909 clauses, which has three models.

In case $\mathscr{F}$ does not even have one model, this means that the considered sets of possible fragment types T and/or possible charges E are not enough to give an interpretation to all considered peaks. If T and E already include all possibilities that should be considered for the current analysis, they cannot be widened. In similar cases, the problem is originated by the presence of uninterpretable non-standard or noise peaks in the spectrum, which may be due to some experimental disturbance in the mass spectrometry analysis. For overcoming this type of problems, the mass spectrometry should be improved. When this option is not available, the formula $\mathscr{F}$ should be considered as an instance of the *maximum satisfiability* problem (Max-SAT) [30], which consists of finding a truth assignment for the variables $x_{j \to t,e}$ maximizing the number of clauses which evaluate to *True*. By doing so, some clauses will stay unsatisfied. Unsatisfied interpretation clauses correspond to a solution not interpreting some peaks, hence having a mismatch number $\nu > 0$. Unsatisfied incompatibility or multicharge clauses mean that not all rules for having a coherent interpretation can be respected in the current analysis. In any of these cases, the result of the analysis is less reliable, but the problem is in the input data.

It is worth to note that the SAT problem, and all its variants above described, can be solved not only working in the field of propositional logic (as it is done by BrChaff and many other solvers), but also working with Integer Linear Programming (ILP). Each clause, written in the following general form (P is the set of indices of positive variables, N that one of negative variables)

$$\bigvee_{k \in P} x_k \vee \bigvee_{k \in N} \neg x_k,$$

can be converted into the following linear inequality

$$\sum_{k \in P} x_k + \sum_{k \in N} (1 - x_k) \geq 1$$

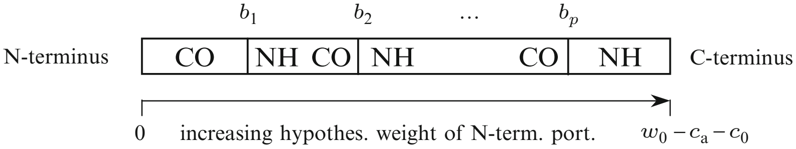
Therefore, the set of all clauses becomes a set of linear inequalities constituting the constraints of the ILP, an objective function can be added, and algorithms for solving ILP can now be used, [21]. Generally speaking, however, the complexity of solving the above described problems does not change: when the SAT problem belongs to an easy special class, the same happens for the ILP. See [10] for further details.

1.6 Computing the Weights Database and Generating the Sequences

As described, each variable $x_{j \to t,e}$ with $j \in F$, $t \in T$, $e \in E$, corresponds to an hypothesized weight $b(j,t,e)$ of an N-terminal portion of the normalized peptide. Therefore, given a model μ for the generated formula $\mathscr{F}$, consider all the hypothesized weights of the N-terminal portions corresponding to all the *True* variables of μ. By ordering such values in increasing weight order, we obtain what we call the *succession of breakpoints* B^{μ} corresponding to model μ for the normalized peptide under analysis.

$$B^{\mu} = \{b_1, b_2, \ldots, b_p\}$$

This means that when giving to the considered peaks W the interpretation represented by μ, we have located the peptidic bonds of the normalized peptide under analysis at the locations given by the values of the elements of B^{μ}, as illustrated by the following diagram.



Definition 1.11. Define now a *gap* as the difference between two adjacent breakpoints (b_{h+1}, b_h), and a corresponding *subsequence* as the portion of the normalized peptide spanning between the two peptidic bonds corresponding to the two above adjacent breakpoints (b_{h+1}, b_h).

Now we compute, for each value of gap $b_{h+1} - b_h$, all the non-negative integer vectors $(y_1, y_2, \dots, y_n)^{tr} \in \mathbb{Z}^n_+$ verifying the following equation.

$$b_{h+1} - b_h = y_1(a_1 - c_a) + y_2(a_2 - c_a) + \cdots + y_n(a_n - c_a) \pm 2\delta$$

The results are all the possible subsequences that may cover the gap $b_{h+1} - b_h$. Denote such set of subsequences by $S(b_{h+1} - b_h)$. Note that $S(b_{h+1} - b_h)$ depends only on the value of the gap $b_{h+1} - b_h$ and not on the locations of the breakpoints. The first gap $b_1 - 0$ and the last one $w_0 - (c_a + c_0) - b_p$ should be managed in a way which is slightly different from that of the central gaps. They are indeed the only gaps that may contain components having limitation on their positions in the sequence (only N-terminal or only C-terminal, see Sect. 1.2); hence, this should be considered. Furthermore, only an imprecision δ instead of 2δ should be considered for the first gap, since only one extremity of the gap can be affected by such imprecision. Define $b_0 = 0$ for a more uniform notation.

In order to compute such subsequences, we use a *weights database* as follows. The possible components of the normalized peptide can be viewed as an alphabet Σ on n symbols. For instance, if the possible components are the 20 amino acids reported in Table 1.1, we have

$$\Sigma = \{\text{Gly}, \text{Ala}, \dots, \text{Tyr}\}$$

A subsequence of the normalized peptide is just a sequence of components and therefore a string over this alphabet. Its weight is normalized and therefore can be computed by summing the weights of the components. The set of all such strings may be denoted as Σ^*. Knowing the correspondences between all the elements of Σ^* and their weights would of course speed-up the operation of finding the subsequences. However, generating all Σ^* would be clearly impossible from a computational point of view. On the other hand, the set of strings having a molecular weight not greater than λ may be denoted as $\Sigma^{*\leq\lambda}$. If λ is greater than or equal to the maximum of the mentioned gaps, also $\Sigma^{*\leq\lambda}$ may give the same help in the operation of finding the subsequences. For useful values of λ, however, $\Sigma^{*\leq\lambda}$ generally becomes too large.

We describe now a procedure to consider it implicitly. Not that $\Sigma^{*\leq\lambda}$, for any fixed λ, can be computed using only the information about the possible components for the current analysis (or better yet, for the set of current analyses). We therefore compute it off-line, before starting any sequence analysis, as soon as the information about the possible components is available. Every sequence is put in correspondence with a natural number, by considering the components of the sequence as a number expressed in base $n + 1$ (n is the number of components). This correspondence must be biunivocal and easily computable. For instance, with the 20 amino acids reported in Table 1.1, considering the sequence written horizontally, the last (the rightmost) element would correspond to the symbol multiplying 21^0, the last-but-one element would correspond to the symbol multiplying 21^1 and so on. Moreover, the first symbol (Gly) in the list of possible components (Table 1.1) would mean number 1, the second (Ala) number 2 and so on. An empty position (no amino acid) would mean number 0. This holds because, if any other amino acid would mean 0, a sequence beginning with that amino acid would correspond to the same number as the same sequence without the initial amino acid, and the correspondence would not be biunivocal.

Example 1.5. The sequence Gly-Ser-Gly-Tyr, or, more precisely,

$$< \text{no amino acid} > \cdots < \text{no amino acid} > \text{Gly Ser Gly Tyr}$$

would then corresponds to the number 0 ... 0 1 3 1 20(or K) in base 21, that in base 10 is $20\times21^0(=20)+1\times21^1(=21)+3\times21^2(=1323)+1\times21^3(=9261)=10625$.

The weights of all sequences up to molecular weight λ are therefore computed off-line and stored in correspondence with the described natural numbers representing the sequences. This computation may be done efficiently using smaller solutions to gradually compute larger solutions. Note that more sequences may have the same molecular weight; hence, one weight may correspond to more than one natural number, even if one natural number corresponds to only one sequence, hence to one weight. The natural numbers may also be not stored, but simply be the indices of an array memorizing the weights. This constitutes the weights database: given a molecular weight, it allows to find almost instantaneously which are all the sequences of components that could produce a portion of normalized peptide having that weight. Value λ is chosen big enough to cover all the possible gaps that one could need to sequence in the set of current analyses.

Therefore, for each gap $b_{h+1}-b_h$, the set of all the possible subsequences $S(b_{h+1}-b_h)$ covering that gap is computed in extremely short times by searching the weights database for all natural numbers corresponding to the weight $b_{h+1}-b_h$, and by explicitly generating the subsequences corresponding to such natural numbers.

When all the sets of subsequences $S(b_{h+1}-b_h),\ h=0,\ldots,p$ are available, all the possible sequences $\mathscr{S}_\mu$ of the normalized peptide under the peak interpretation μ can be generated with the concatenation of such sets in all possible ways, operation which we denote by $\oplus$, but eliminating sequences violating the requirements regarding minimum m_i or maximum M_i value on the number of each component.

$$\mathscr{S}_\mu = S(b_1-b_0)\oplus S(b_2-b_1)\oplus\cdots\oplus S(w_0-c_a-c_0-b_p)$$

Finally, when considering the sets of all the possible sequences $\{\mathscr{S}_{\mu_1},\mathscr{S}_{\mu_2},\ldots,\mathscr{S}_{\mu_r}\}$ for all the possible models $\{\mu_1,\mu_2,\ldots,\mu_r\}$ of $\mathscr{F}$, the complete set of all possible sequences $\mathscr{S}$ of the normalized peptide is obtained:

$$\mathscr{S} = \mathscr{S}_{\mu_1}\cup\mathscr{S}_{\mu_2}\cup\cdots\cup\mathscr{S}_{\mu_r}$$

By construction, the set of all the possible sequences $\mathscr{S}$ of the normalized peptide is also the set of all the possible sequences of the real peptide under analysis; hence, the sequencing problem have been solved.

Note that, in the case when the formula $\mathscr{F}$ is unsatisfiable, and a truth assignment maximizing the number of clauses which evaluates to *True* has been found, some gap may admit no subsequences because some incompatibility clauses are not respected. A less reliable solution can in this case be obtained by merging each unsequenceable gap with one of its neighboring ones (preferably the smaller).

Example 1.6. When considering the formula $\mathscr{F}$ of Example 1.4 with 108 variables, 4909 clauses and three models, computing the weights database with $\lambda = 300$ we obtain three breakpoint successions, reported below together with all their corresponding possible sequences:

{87.0, 224.2, 339.2, 452.2, 565.2, 662.2} which gives two sequences:
Ser-His-Asp-Leu-Leu-Pro-Gly-Leu
Ser-His-Asp-Leu-Leu-Pro-Leu-Gly

{87.0, 224.2, 339.2, 452.2, 565.2, 678.3} which gives two sequences:
Ser-His-Asp-Leu-Leu-Leu-Gly-Pro
Ser-His-Asp-Leu-Leu-Leu-Pro-Gly

{87.0, 184.0, 355.2, 452.2, 565.2, 662.2} which gives four sequences:
Ser-Pro-Gly-Asn-Pro-Leu-Pro-Gly-Leu
Ser-Pro-Gly-Asn-Pro-Leu-Pro-Leu-Gly
Ser-Pro-Asn-Gly-Pro-Leu-Pro-Gly-Leu
Ser-Pro-Asn-Gly-Pro-Leu-Pro-Leu-Gly

However, since in this series of examples we selected from the spectrum of Fig. 1.1 only the labelled peaks, results are not as accurate as it would be possible when selecting more peaks.

1.7 Implementation and Results

The described approach is implemented in C++. The initial input routine (1) reads all informations about possible components and possible types of fragments and charges and computes the weights database, and (2) reads the spectrum and extracts from it all peaks above a certain value. After this, the logic formula $\mathscr{F}$ representing the peak interpretation problem is generated. All models of $\mathscr{F}$ are then found by means of the DPLL SAT solver BrChaff [6], modified in order to search for all the models of the given formula. Then, for each model μ of $\mathscr{F}$, the breakpoint succession is computed, and all the possible subsequences covering each gap are computed and linked together.

Those subsequences may be produced either by means of a specialized branching algorithm working on-line, or by means of the weights database computed off-line and used on-line. Finally, by considering the union of the set of sequences corresponding to the different models of $\mathscr{F}$, all the solutions of the sequencing problem are obtained.

Table 1.2 Real-world peptide sequencing problems

	Input data				Outcomes				Times	
w_0	f	$t_{\max}$	$e_{\max}$	n	x	v	r	$\mathscr{S}$	w/o WD	w WD
572.20	7	2	1	20	14	108	1	1	0.1	0.1
572.20	7	6	2	20	84	3,571	2	2	1.9	1.6
851.30	18	2	1	20	36	543	1	4	0.5	0.5
851.30	18	4	2	24	144	6,780	4	7	2.0	1.4
851.30	18	6	3	24	324	12,642	10	16	5.6	3.0
859.12	20	3	1	40	60	2,904	4	26	1.6	1.1
859.12	20	6	2	40	240	8,156	5	29	4.1	3.4
913.30	16	2	1	20	32	539	2	7	1.0	0.8
913.30	16	6	3	20	288	10,741	8	32	6.8	4.0
968.58	19	2	1	20	38	768	6	24	1.3	1.1
968.58	19	6	2	20	228	7,021	10	38	4.1	3.4
1037.10	18	2	1	20	36	714	7	25	1.4	1.0
1037.10	18	6	2	20	216	6,936	12	44	4.3	3.2
1108.60	21	2	1	26	42	2,687	8	18	3.5	2.1
1108.60	21	4	2	26	168	7,456	16	64	12.2	5.6
1234.20	19	2	2	20	76	4,529	9	26	8.3	3.2
1234.20	19	6	2	20	228	8,956	15	106	29.2	14.0
1479.84	20	2	1	20	40	690	7	22	14.3	6.8
1479.84	20	6	2	20	240	8,796	18	102	33.9	13.7
1570.60	22	2	1	21	44	2,498	9	35	28.5	16.3
1570.60	22	6	2	21	264	9,657	14	98	56.8	39.2
1607.69	27	2	2	26	108	5,744	6	20	44.3	20.9
1607.69	27	6	3	26	486	22,565	11	63	473.0	192.8

Table 1.2 reports various experiments of real peptide sequencing problems on a Pentium IV 1.7GHz PC. In particular, we indicate: the weight of the peptide (w_0); the number of peaks extracted from the spectrum (f); the number of considered types ($t_{\max}$) and charges ($e_{\max}$) of fragments; the number of possible components (n); the number of variables (x) and clauses (v) of the obtained formula; the number of models (r) of the obtained formula, the overall number of solutions ($\mathscr{S}$), and computational times (in seconds) for the whole sequencing procedure without the weights database (w/o WD) and with it (w WD). Time for computing off-line the weights database with $\lambda = 300$ is 40 seconds and with $\lambda = 400$ is 126 seconds. Both values were sufficient for sequencing the gaps in the reported analyses. A time of this order (the exact one depends on our a priori choice for λ) should therefore be considered just once for a whole series of tests with WD. It can also be stored on hard disk and read by the input routine in a subsequent time. Those results are intended to give real-world examples of application, rather than exploring all the computational possibilities of the described procedure.

As observable from the table, results depend of course on the choice of possible types and charges of fragments: for the same spectrum, different choices produce different results, and the number of sequences compatible with the given input data is sometimes large. This is an intrinsical character of the problem. However, all the solutions are generally very related, in the sense that some parts are just common, and some other are given by all the combinations of a (generally small) number of components.

The use of the weights database is always able to reduce computation times. This reduction increases when increasing the solution time and grows faster than the latter one. In the examples, it passes from about 0.2 seconds for a problem with solution time of 1 second, i.e., a reduction of 20%, to about 280 seconds for a problem with solution time of 473 seconds, i.e., a reduction of 59%. Therefore, the more consistent speed-ups are obtained for the larger instances (the ones for which they are more useful). The whole procedure is a powerful, accurate and flexible sequencing tool, and allows the sequencing of compounds not handled by other available techniques.

1.8 Conclusions

The problem of the determination of the amino acid sequence of a peptide is considered. Such problem is of basic relevance in biological and medical research, but is difficult to model and computationally hard to solve. Data obtained from the mass spectrometry analysis of a generic polymeric compound, constituted, according to specific chemical rules, by a sequence of components, are here used to build a propositional logic formula. The models of this formula represent coherent interpretations of the set of data and are used to generate all possible correct results of the analysis itself. The problem has been therefore subdivided into a *peaks interpretation* phase and a *sequence generation* phase. The peaks interpretation phase is solved by means of a DPLL SAT solver modified in order to search for all the models of a formula. The sequence generation phase is solved by computing off-line a weights database, so that all sequences up to a certain molecular weight can be considered implicitly, but only the needed ones generated explicitly. Results of tests on real-world peptide sequencing problems demonstrate the effectiveness of this approach. The use of the weights database is able to sensibly reduce computation times, especially for larger instances.

Acknowledgements The author is grateful to Prof. G. Gianfranceschi and to Prof. G. Koch for stimulating discussions and for providing details on the biological aspects of the problems. Partially reprinted from [5] with permission from Elsevier. Italian Patent number: MI2002A000396. International Patent Application number: PCT/IB03/00714.

References

1. B. Aspvall, M.F. Plass, and R.E. Tarjan. A linear time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters* 8, 121–123 (1979)
2. V. Bafna and N. Edwards. On de novo interpretation of tandem mass spectra for peptide identification. In *Annual Conference on Research in Computational Molecular Biology* RECOMB03, 9–18 (2003)

3. E. Boros, Y. Crama, and P.L. Hammer. Polynomial time inference of all valid implications for Horn and related formulae. *Annals of Mathematics and Artificial Intelligence* 1, 21–32 (1990)
4. C. Branden and J. Tooze. *Introduction to Protein Structure.* Garland Publishing, New York (1999)
5. R. Bruni. Solving peptide sequencing as satisfiability. *Computer and Mathematics with Applications* 55(5), 912–923 (2008)
6. R. Bruni and A. Santori. Adding a new conflict-based branching heuristic in two evolved DPLL SAT solvers. In *Proceedings of the Seventh International Conference on Theory and Applications of Satisfiability Testing* SAT2004 (2004)
7. R. Bruni, G. Gianfranceschi, and G. Koch. On peptide de novo sequencing: a new approach. *Journal of Peptide Science* 11, 225–234 (2005)
8. G. Casella and C.P. Robert. *Monte Carlo Statistical Methods*. Springer, New York (2006)
9. V. Chandru and J.N. Hooker. Extend Horn clauses in propositional logic. *Journal of the ACM* 38, 203–221 (1991)
10. V. Chandru and J.N. Hooker. *Optimization Methods for Logical Inference*. Wiley, New York (1999)
11. T. Chen, M.Y. Kao, M. Tepel, J. Rush, and G.M. Church. A dynamic programming approach to de novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology* 8(6), 571–583 (2001)
12. W.F. Clocksin. Logic programming and digital circuit analysis. *Journal of Logic Programming* 4(1), 59–82 (1987)
13. M. Conforti and G. Cornuéjols. A class of logical inference problems soluble by linear programming. *Journal of the ACM* 42(5), 1107–1113 (1995)
14. V. Dancik, T.A. Addona, K.R. Clauser, J.E. Vath, and P.A. Pevzner. De novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology* 6, 327–342 (1999)
15. M.R. Garey and D.S. Johnson. *Computers and Intractability*. Freeman, New York (1979)
16. J. Gu, P.W. Purdom, J. Franco, and B.W. Wah. Algorithms for the Satisfiability (SAT) Problem: A Survey. *DIMACS Series in Discrete Mathematics* 35, 19–151, American Mathematical Society (1997)
17. R.S. Johnson and J.A. Taylor. Searching sequence databases via de novo peptide sequencing by tandem mass spectrometry. *Methods in Molecular Biology* 146, 41–61 (2000)
18. H. Kleine Büning and T. Lettman. *Propositional Logic: Deduction and Algorithms*. Cambridge University Press, Cambridge (1999)
19. T.D. Lee. Fast atom bombardment and secondary ion mass spectrometry of peptides and proteins. In *Methods of Protein Microcharacterization*, J.E. Shively (editor) 403–441, Humana Press, NJ (1986)
20. G. Montaudo and R.P. Lattimer (editors). *Mass Spectrometry of Polymers*. CRC Press, Boca Raton (2001)
21. G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* Wiley, New York (1988)
22. J.S. Schlipf, F.S. Annexstein, J.V. Franco, and R.P. Swaminathan. On finding solutions for extended horn formulas. *Information Processing Letters* 54(3), 133–137 (1995)
23. G. Siuzdak. *Mass Spectrometry for Biotechnology*. Academic Press, New York (1996)
24. Software system DeNovoX. ThermoFinnigan Corp. (http://www.thermo.com)
25. Software system Mass Seq. Micromass Ltd. (http://www.micromass.co.uk)
26. Software system PEAKS. Bioinformatics Solutions Inc. (http://www.bioinformaticssolutions.com)
27. Software system Spectrum Mill. Agilent Technologies Inc. (http://www.agilent.com)
28. J.T. Stults. Peptide sequencing by mass spectrometry. *Methods of Biochemical Analysis* 34, 145–201 (1990)
29. J.A. Taylor and R.S. Johnson. Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Analytical Chemistry* 73, 2594–2604 (2001)
30. K. Truemper. *Effective Logic Computation*. Wiley, New York (1998)
31. P. Van Hentenryck. *Constraint satisfaction in logic programming.* MIT, MA (1989)
32. D. Voet. *Biochemistry*. Wiley, New York (2004)

Chapter 2
Divide and Conquer Strategies for Protein Structure Prediction

Pietro Di Lena, Piero Fariselli, Luciano Margara, Marco Vassura, and Rita Casadio

Abstract In this chapter, we discuss some approaches to the problem of protein structure prediction by addressing "simpler" sub-problems. The rationale behind this strategy is to develop methods for predicting some interesting structural characteristics of the protein, which can be useful per se and, at the same time, can be of help in solving the main problem. In particular, we discuss the problem of predicting the protein secondary structure, which is at the moment one of the most successful sub-problems addressed in computational biology. Available secondary structure predictors are very reliable and can be routinely used for annotating new genomes or as input for other more complex prediction tasks, such as remote homology detection and functional assignments. As a second example, we also discuss the problem of predicting residue–residue contacts in proteins. In this case, the task is much more complex than secondary structure prediction, and no satisfactory results have been achieved so far. Differently from the secondary structure sub-problem, the residue–residue contact sub-problem is not intrinsically simpler than the prediction of the protein structure, since a roughly correctly predicted set of residue–residue contacts would directly lead to prediction of a protein backbone very close to the real structure. These two protein structure sub-problems are discussed in the light of the current evaluation of the performance that are based on periodical blind-checks (CASP meetings) and permanent evaluation (EVA servers).

2.1 Introduction

Methods developed for the problem of the protein structure prediction in silico aim at finding the three-dimensional (3D) conformation of the protein starting from its amino-acidic residue sequence (primary structure) [7]. Protein function is strictly dependent on the native protein 3D structure and protein structure prediction is one

P. Di Lena (✉)
Department of Computer Science, University of Bologna, Italy
e-mail: dilena@cs.unibo.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_2,

of the most important and mostly studied problems of computational biology [26]. Despite many efforts, an acceptable solution for new sequences, not having homologous sequences for which the 3D structure is known, is still to be found. Given the difficulty to compute directly the protein 3D structure, many intermediate problems have been addressed. One way to simplify the problem is to compute features that are local with respect to the backbone of the protein. These are called secondary structure motifs and are well characterised as alpha-helices, beta-sheets and coil on the basis of specific values of torsion angles. The problem of predicting secondary structures in proteins has been also addressed with machine learning methods and it is presently considered one of the most successful problems of computational biology [43]. In this chapter, we will comment on the most successful implementations of protein secondary structure prediction methods.

However, even when well predicted, secondary structure alone does not carry enough information to understand protein 3D conformation. To this aim, it would suffice to find global distance constraints between each couple of residues. This sub-problem is commonly known as residue–residue contact prediction and it has been again addressed with machine learning methods [3]. Residue–residue contact prediction is today the only method that can grasp in a simplified manner long-range interactions between residues of a protein sequence. Although the problem is still far from being solved, we will review the most efficient algorithms that are presently the state-of-the-art methods in the field.

So far, the most interesting results in secondary structure prediction and residue–residue contact prediction have been achieved by a clever combination of machine-learning methods with evolutionary information available in the ever growing databases of protein structures [1, 11, 18, 20].

In order to make the chapter self-contained as much as possible, in the following sections we briefly review the most basic concepts of machine learning methods (Sect. 2.2) and the most commonly used techniques for extracting evolutionary information from databases of protein sequences (Sect. 2.3). The rest of the chapter is devoted to the detailed description of the most famous secondary structure predictors (Sect. 2.4) and residue–residue contact predictors (Sect. 2.5). For both topics, we also describe in detail the standard evaluation criteria adopted to measure the performance of the predictors and outline what is the state of the art in terms of the respective evaluation criteria according to the experiments performed at CASP meetings[1] and EVA server.[2]

2.2 Data Classification with Machine Learning Methods

Machine learning is concerned with the design and development of algorithms for the acquisition and integration of knowledge. Biological data classification is a typical problem usually approached with machine learning methods.

[1] http://predictioncenter.org/

[2] http://cubic.bioc.columbia.edu/eva/

Data classification is the problem of assigning objects to one of the mutually exclusive classes according to statistical properties derived from a training set of examples sharing the same nature of such objects. The problem can be easily formalised in the following way. Assume that the data we want to classify is represented by a set of n-dimensional vectors $x \in X = \mathbb{R}^n$ and that each one of such vectors can be assigned to exactly one of m possible classes $c \in C = \{1, \ldots, m\}$. Given a set of pre-compiled examples $E = \{(x_1, c_1), \ldots, (x_k, c_k)\}$, where $(x_i, c_i) \in X \times C$ and $|E| < |X|$, the objective is to learn from E a mapping $f : X \rightarrow C$ that assigns every $x \in X$ to its correct class $c \in C$. In the biological context, each entry of the vector $x \in X$ usually represents a single feature (observation) of the object we want to classify (i.e., x is not the object itself), and the number of classes is typically limited to two/three. Moreover, machine learning methods generally do not provide a rigid classification of an object; they instead return the probability that the object belongs to each one of the possible classes (a classification can be obtained by choosing the class with higher probability). In bioinformatics, the most widely used machine learning methods for data classification are neural networks (NN), support vector machines (SVM) and Hidden Markov models (HMM). We do not discuss here the features and the limitations of such methods (for an extensive introduction, see [5]), but we briefly outline the problem of correctly evaluating the performance of predictors of protein structural characteristic.

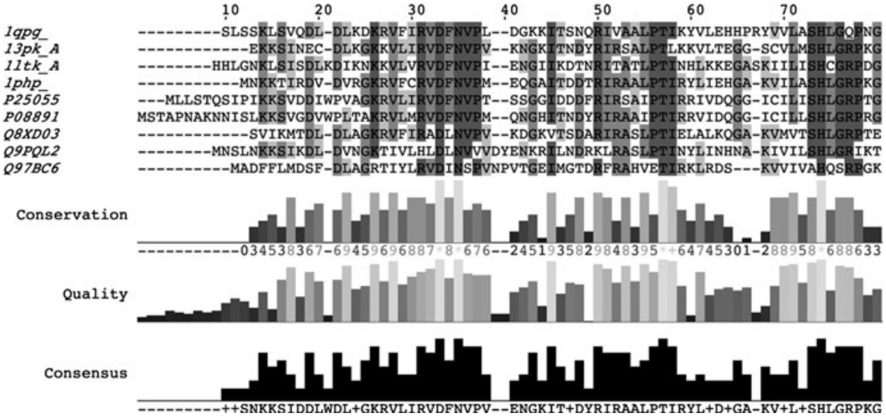
A reliable approach for assessing the performance of data classification is a necessary pre-condition for every machine learning-based method. The cross-validation is the standard technique used to statistically evaluate how accurate a predictive model is. The cross-validation involves the partitioning of the example set into several disjoint sets. In one round of cross-validation, one set is chosen as test set and the others are used as training set. The method is trained on the training set and the statistical evaluation of the performance is computed from the prediction results obtained on the test set. To reduce variability, multiple cross-validation rounds are performed by interchanging training and test sets, and the results obtained are averaged over the number of rounds.

A proper evaluation (or cross-validation) of prediction methods needs to meet one fundamental requirement: the test set must not contain examples too much similar to those contained in the training set. When testing prediction methods for protein features (such as secondary structure or inter-residue contacts), this requirement transduces in having test and training sets compiled from proteins that share no significant pairwise sequence identity (typically <25%). If homologous sequences are included in both training and test set, the average prediction accuracy does not provide a reliable estimation of the performance, and, in particular, it does not reflect the performance of the method for sequences not homologue to those in the training set.

2.3 Evolutionary Information and Multiple Sequence Alignments

One of the most successful tools in bioinformatics is the introduction of evolutionary information as a key ingredient for protein structure and function predictions. The evolutionary information contained in a set of (related) protein sequences can be extracted from a multiple alignment of all the sequences in the set. The multiple sequence alignment (MSA) refers to the problem of aligning three or more sequences in order to identify their regions of similarity. An MSA of a set of protein sequences is represented as a matrix, where each row corresponds to a single sequence and each column corresponds to a set of aligned residues, one for each protein in the set (Fig. 2.1).[3,4] When properly computed, each column of the MSA encodes the possible evolutionary mutations that can occur at the corresponding positions in the sequences included in the MSA. Those columns of the MSA that exhibit low variability correspond to regions that are highly conserved with respect to the evolutionary mutations of protein sequences.

In a pioneering work, Benner and Gerloff [4] introduced the idea that multiple sequence alignments can improve protein structure prediction. Their basic concept relies on the fact that the most conserved regions of a protein sequence (in terms of multiple alignments) are those regions which are either functionally important, and/or buried in the protein core. By this, Benner and Gerloff demonstrated that the degree of solvent accessibility of an amino acid residue could be predicted with



```
                  10        20        30        40        50        60        70
1qpg_     ---------SLSSKLSVQDL-DLKDKRVFIRVDFNVPL--DGKKITSNQRIVAALPTIKYVLEHHPRYVVLASHLGQPNG
13pk_A    ------------EKKSINEC-DLKGKKVLIRVDFNVPV--KNGKITNDYRIRSALPTLKKVLTEGG-SCVLMSHLGRPKG
11tk_A    --------HHLGNKLSISDLKDIKNKKVLVRVDFNVPI--ENGIIKDTNRITATLPTINHLKKEGASKIILISHCGRPDG
1php_     -----------MNKKTIRDV-DVRGKRVFCRVDFNVPM--EQGAITDDTRIRAALPTIRYLIEHGA-KVILASHLGRPKG
P25055    ---MLLSTQSIPIKKSVDDIWPVAGKRVLIRVDFNVPT--SSGGIDDDFRIRSAIPTIRRIVDQGG-ICILISHLGRPTG
P08891    MSTAPNAKNNISLKKSVGDVWPLTAKRVLMRVDFNVPM--QNGHITNDYRIRAAIPTIRRVIDQGG-ICILLSHLGRPRG
Q8XD03    ------------SVIKMTDL-DLAGKRVFIRADLNVPV--KDGKVTSDARIRASLPTIELALKQGA-KVMVTSHLGRPTE
Q9PQL2    --------MNSLNKKSIKDL-DVNGKTIVLHLDLNVVVDYENKRILNDRKLRASLPTINYLINHNA-KIVILSHLGRIKT
Q97BC6    ----------MADFFLMDSF-DLAGRTIYLRVDINSPVNPVTGEIMGTDRFRAHVETIRKLRDS---KVVIVAHQSRPGK

Conservation
          ------------034538367-69459696887*8*676--24519358298483 95*+64745301-288958*688633

Quality

Consensus
          ---------++SNKKSIDDLWDL+GKRVLIRVDFNVPV--ENGKIT+DYRIRAALPTIRYL+D+GA-KV+L+SHLGRPKG
```

Fig. 2.1 Multiple sequence alignment taken from the BAliBASE3 database (example BB50004 from RV50 reference set) and visualised with the Jalview software. Only the first 80 positions of the alignment are visualised. The symbol "–" denotes a gap. Darker columns of the MSA correspond to higher conserved regions. The only perfectly conserved positions are 33, 35, 57 and 74

[3] BAliBASE3 database: http://www-bio3d-igbmc.u-strasbg.fr/balibase/

[4] Jalview software: http://www.jalview.org/

reasonable accuracy by clustering the sequences in an aligned family, and assessing the degree of sequence variability observed between very similar pairs. Lately, this idea was exploited by Rost and Sander, who showed that it was possible to improve the accuracy of the prediction of secondary structures and solvent accessibility introducing evolutionary information in the form of sequence profiles as input to neural networks [42].

Differently from an MSA, whose dimension increases linearly with the number of aligned sequences, a sequence profile of a protein is a matrix P whose columns represent the sequence positions and whose rows are the 20 possible residue symbols. The profile matrix P is computed from a MSA and it is relative to a specific sequence of interest p. Each element P_{ai} of the sequence profile represents the normalised frequency of the residue type a in the aligned position i. In practice, given an MSA that contains the sequence of interest p, we derive the column i of the corresponding profile by computing the frequencies of occurrence of each residue in the column of the MSA corresponding to the ith residue of p. In this way, the information contained in a profile P is not dependent on the number of aligned sequences so that it becomes easy to use fragments of the matrix P as input for machine learning methods.

The computation of an MSA for a query sequence is a complex process both in terms of time and care required. It consists of two steps. First, a search of the query sequence against a non-redundant dataset of protein sequences is needed in order to select a set of chains that are similar to the query one. There are several optimal and near-optimal pairwise-alignment algorithms to perform such searches. Currently, the heuristic basic local alignment search tool (BLAST) [2] is considered the standard-de-facto software for pairwise sequence comparison. Despite the fact that exact algorithms are available for pairwise sequence comparison, the heuristic BLAST is the most widely used due to its speed (non-redundant datasets can contain millions of different protein sequences) and good performance compared to exact algorithms. The selection of similar sequences must be performed carefully in order to avoid the introduction of meaningless sequences in the MSA, such as sequences with low complexity regions. Low complexity regions represent sequences of very non-random composition (“simple sequences,” “compositionally-biased regions”). They are abundant in natural sequences and may determine high scoring matching segments in unrelated protein sequences. To avoid this problem, BLAST implements a filter procedure based on the SEG [49] software. SEG provides a measure of compositional complexity of a sequence segment and divides sequences into contrasting segments of low complexity and high complexity. Typically, globular domains have higher sequence complexity than fibrillar or conformationally disordered protein segments. When used in BLAST, SEG replaces the low complexity regions within the input sequence with X’s to prevent spurious matching with unrelated sequences.

When the set of similar sequences has been selected, the second step consists of building an MSA. Differently from the pairwise sequence alignment problem, building an optimal multiple alignment is a difficult task and it is not computable in reasonable time. Several software implementations of heuristic algorithms for MSA are available (MAXHOM [44], CLUSTALW [47], T-Cofee [29] and MUSCLE [10] are currently the most widely used) and none of them is globally accepted as a standard.

Few years ago, the procedure described above for building MSA was almost standard and time-consuming; thus, during the construction and tuning of new prediction methods most of the researchers used the homology-derived secondary structure of proteins (HSSP) precompiled multiple sequence alignments generated with the MAXHOM software. Currently, a faster and more accurate method for the construction of reliable sequence profiles is the adoption of the position specific iterative (PSI) feature in BLAST [2]. In PSI-BLAST a sequence profile and a position-specific scoring matrix (PSSM) are automatically constructed from a pseudo-multiple alignment of the highest scoring hits in an initial BLAST search. The PSSM is generated by calculating position-specific scores for each position in the alignment. Highly conserved positions receive high scores and weakly conserved positions receive scores near zero. The profile is used to perform a further BLAST search and the current profile is refined according to the outcomes of the new search. This iterative procedure is performed until the retrieved sequences remain constant or a fixed number of iterations are achieved. In [21], the prediction accuracy of secondary structure was improved by using directly the PSI-BLAST PSSM to feed a neural network system.

2.4 Secondary Structure Prediction

In biochemistry and structural biology, the protein secondary structure refers to the three-dimensional shape of consecutive residue segments. The most common secondary structure elements are alpha-helices and beta-sheets. The formation of secondary structure elements is mostly guided by local inter-residue interactions mediated by hydrogen bonds. For example, an alpha-helix is formed when hydrogen bonds occur regularly between positions i and $i+4$ in a protein segment. When hydrogen bonds occur between positions i and $i + 3$, then a 3_{10} helix is formed. A beta-sheet is formed when two strands are joined by hydrogen bonds involving alternating residues on each participating strand. In the 1950s, Pauling correctly guessed the formation of helices and strands [31, 32], before any protein structure had been determined experimentally.

There are several methods for defining protein secondary structure elements. The dictionary of protein secondary structure (DSSP) method [23] is actually considered the de facto standard for secondary structure definition. The DSSP defines eight types of secondary structure elements, based on hydrogen-bonding patterns as those initially proposed by Pauling (Fig. 2.2):

- G = 3-turn helix (3_{10} helix). Min length three residues.
- H = 4-turn helix (alpha helix). Min length four residues.
- I = 5-turn helix (pi helix). Min length five residues.
- T = hydrogen bonded turn (3, 4 or 5 turn).

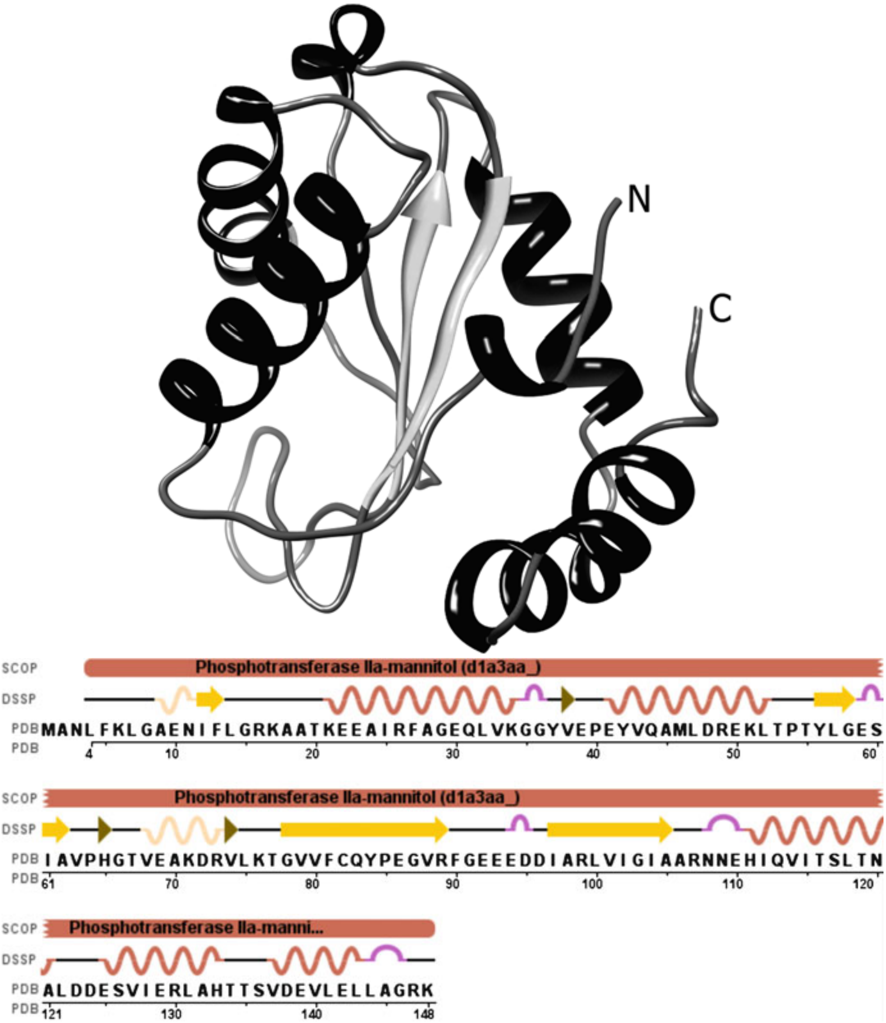

N
C
SCOP
Phosphotransferase IIa-mannitol (d1a3aa_)
DSSP
PDB MANLFKLGAENIFLGRKAATKEEAIRFAGEQLVKGGYVEPEYVQAMLDREKLTPTYLGES
PDB 4 10 20 30 40 50 60
SCOP
Phosphotransferase IIa-mannitol (d1a3aa_)
DSSP
PDB IAVPHGTVEAKDRVLKTGVVFCQYPEGVRFGEEEDDIARLVIGIAARNNEHIQVITSLTN
PDB 61 70 80 90 100 110 120
SCOP
Phosphotransferase IIa-manni...
DSSP
PDB ALDDESVIERLAHTTSVDEVLELLAGRK
PDB 121 130 140 148

Fig. 2.2 Graphical representation of the *Escherichia coli* phosphotransferase IIAmannitol (1a3a chain A, 148 residues). The figure above shows the three-dimensional structure, highlighting helices, strands (*arrows*) and coils (*irregular loops*). The figure below shows the amino-acidic sequence and the respective DSSP secondary structure elements

- E = extended strand in parallel and/or anti-parallel beta sheet conformation. Min length two residues.
- B = residue in isolated beta bridge (single pair beta-sheet hydrogen bond formation).
- S = bend (the only non-hydrogen bond-based assignment).
- C = every residue that cannot be assigned to any of the above conformations.

It is worth noting that the eight-state DSSP vocabulary is just a simplification of the possible variations of hydrogen-bonding patterns present in proteins. For example, the class C stands for loops or irregular elements, which are often called coils or random coils. In order to simplify the DSSP classification, most of the secondary structure prediction methods reduce further the DSSP vocabulary into three most characteristic states, helix (H), strand (E) and other (L), according to the scheme proposed in the secondary structure section of EVA server (EVAsec[5]): H includes (H,G,I), E includes (E,B) and L includes all the others.

Predicting the protein tertiary structure from only its amino acid sequence is actually one of the most challenging problems in structural bioinformatics. In contrast, the secondary structure prediction is more tractable and has been successfully addressed in the last decades. In particular, the successful results in this field have been achieved by combining machine learning methods with evolutionary information available in the ever-growing databases of protein structures. Early secondary structure prediction methods were based on statistics derived from protein segments [6, 16]. The statistics were used to predict how likely the central residue in the segment is in some particular secondary structure element. Several different methods (machine learning based and not) were exploited to derive statistics from protein segments. The accuracy of all these methods was limited to slightly more than 60%. A first significant step-forward in prediction accuracy was made by exploiting evolutionary information encoded in MSA [43]. The PHD predictor by Rost and Sander [42] is the first method that used MSA successfully for secondary structure prediction and that was able to achieve a prediction accuracy $>$70%. The next step-forward was made using more accurate evolutionary information resulting from improved searches and larger databases. The PSIpred method by Jones [21] is historically the first method for secondary structure prediction that cleverly used position-specific alignments from PSI-BLAST and that achieved a further improvement of slightly more than 5% in accuracy. The accuracy of modern secondary structure prediction methods is currently about 77%. While this is not the best possible we can do, due to the approximations made by the DSSP in assigning secondary structure classes, the theoretical limit of prediction accuracy has been estimated approximately 88% [41].

The performances of secondary structure prediction methods were evaluated in CASP1 experiments from CASP1 (1994) to CASP5 (2002). Starting form CASP6, the secondary structure prediction category was not included in the experiments since the progress in this area was too little to be detected with the few amounts of data available in CASP sessions. Currently, larger scale benchmarking is continuously assessed by the EVAsec experiments. In the following section (Sect. 2.4.1), we review the most important measures of secondary structure prediction accuracy (as defined in EVAsec) and we also provide a comparison of some secondary structure prediction methods in terms of these measures. We conclude (Sect. 2.4.2) with the detailed description of two secondary structure predictors, PHD (Sect. 2.4.2.1) and PSIpred (Sect. 2.4.2.2).

[5] http://cubic.bioc.columbia.edu/eva/doc/intro_sec.html

2.4.1 EVAsec: Evaluation of Secondary Structure Prediction Servers

The objectives of EVAsec[3] are to provide a continuous, fully automated and statistically significant analysis of protein secondary structure prediction servers. EVAsec continuously evaluates *secondary structure prediction servers* in real time, whenever new data are available. Secondary structure prediction servers are fully automated websites that accept prediction tasks on request and provide answers in electronic format. At the moment (April 2009), EVAsec is running since 303 weeks and monitors 13 servers.

The most simple and widely used measure of secondary structure prediction accuracy used in EVAsec is the *per-residue prediction accuracy*:

$$Q_3 = 100 \cdot \frac{1}{N} \sum_{i=1}^{3} M_{ii}, \tag{2.1}$$

where N is the length of the protein and $M \in \mathbb{N}^{3\times 3}$ is the confusion matrix, i.e., M_{ij} is equal to the number of residues observed in state i and predicted in state j with $i, j \in \{H, E, L\}$. Since a typical protein contains about 32% H, 21% E, 47% L, the correct prediction of class L tends to dominate the overall accuracy. There are several other measures defined in EVAsec (such as per-state/per-segment accuracy) that can be used to limit this effect. The per-state measures are based on the *Matthews correlation coefficient*:

$$C_i = \frac{p_i \cdot n_i - u_i \cdot o_i}{\sqrt{(p_i + u_i)\cdot(p_i + o_i)\cdot(n_i + u_i)\cdot(n_i + o_i)}}, \tag{2.2}$$

where $i \in \{H, E, L\}$, $p_i = M_{ii}$ (true positives), $n_i = \sum_{j\neq i}^{3} \sum_{k\neq i}^{3} M_{jk}$ (true negatives), $o_i = \sum_{j\neq i}^{3} M_{ji}$ (false positives) and $u_i = \sum_{j\neq i}^{3} M_{ij}$ (false negatives). The most important per-segment accuracy is the *Segment OVerlap* (SOV) measure, based on the average segment overlap between the observed and predicted segment instead of the average per-residue accuracy:

$$\mathrm{SOV} = \frac{100}{N} \sum_{(s_1,s_2)\in S} \frac{\mathrm{minOV}(s_1,s_2) + \delta(s_1,s_2)}{\mathrm{maxOV}(s_1,s_2)} \cdot \mathrm{len}(s_1), \tag{2.3}$$

where

- s_1, s_2 are, respectively, the observed and predicted secondary structure segments in state $i \in \{H, E, L\}$, i.e., all residues of s_1, s_2 are in state i,
- S is the set of segment pairs (s_1, s_2) that are both in the same state i and that overlap at least by one residue. Conversely, S' is the set of observed segments s_1 for which there is no predicted overlapping segment s_2.
- $\mathrm{len}(s_1)$ is the number of residues in segment s_1,
- $N = \sum_{(s_1,s_2)\in S} \mathrm{len}(s_1) + \sum_{s_1\in S'} \mathrm{len}(s_1)$ is the normalization value,

- minOV(s_1, s_2) is the length of actual overlap of s_1 and s_2, i.e., the extent for which both segments have residues in state i,
- maxOV(s_1, s_2) is the length of the total extent for which either of the segments s_1 or s_2 has a residue in state i,
- $\delta(s_1, s_2)$ is equal to

$$\min \left\{ \begin{array}{l} \mathrm{maxOV}(s_1, s_2) - \mathrm{minOV}(s_1, s_2), \mathrm{minOV}(s_1, s_2), \\ \mathrm{int}(\mathrm{len}(s_1)/2), \mathrm{int}(\mathrm{len}(s_2)/2) \end{array} \right\}$$

The accuracy of prediction of the 13 servers currently monitored by EVAsec is given in Table 2.1. The second column of the table gives the number of proteins predicted by each method and the third column gives the average accuracy (Q_3) over all proteins for the respective method. The results in Table 2.1 cannot be used for comparison, since different sets of proteins are used for each method. In Table 2.2, six methods are compared on their largest common subset of 80 proteins. In Table 2.2, also SOV and per-state accuracy measures C_H, C_E, C_L are included.

Table 2.1 Average prediction accuracy (third column) for each secondary structure server monitored in EVAsec (data updated at April 2009). Different sets of proteins are used for each method (the number of proteins used is given in the second column)

Method	Num. proteins	Q_3
APSSP2 [39]	122	75.5
PHDpsi [37]	229	75.0
Porter [33]	73	80.0
PROF_king [30]	230	72.1
PROFsec [40]	232	76.6
PSIpred [21]	224	77.9
SABLE [36]	232	76.1
SABLE2 [36]	159	76.8
SAM-T99sec [24]	204	77.3
SCRATCH (SSpro3) [34]	207	76.2
SSpro4 [34]	144	77.9
Yaspin [27]	157	73.6

Table 2.2 Performance comparison of six secondary structure prediction methods on their largest common subset of 80 proteins as evaluated in EVAsec (data updated at April 2009). The average of three different accuracy measures ± standard deviation are given: Q_3 (see (2.1)), SOV (see (2.3)) and C_H, C_E, C_L (see (2.2)). The first column of the table gives the rank of the corresponding predictor

Rank	Method	Q_3	SOV	C_H	C_E	C_L
1	PROFsec	75.5 ± 1.4	74.9 ± 1.9	0.65 ± 0.03	0.70 ± 0.04	0.56 ± 0.02
	PSIpred	76.8 ± 1.4	75.4 ± 2.0	0.67 ± 0.03	0.73 ± 0.04	0.55 ± 0.02
	SAM-T99sec	77.2 ± 1.2	74.6 ± 1.5	0.67 ± 0.03	0.71 ± 0.03	0.59 ± 0.02
2	PHDpsi	73.4 ± 1.4	69.5 ± 1.9	0.64 ± 0.03	0.68 ± 0.04	0.52 ± 0.02
3	PROF_king	71.6 ± 1.5	67.7 ± 2.0	0.62 ± 0.03	0.68 ± 0.04	0.51 ± 0.02

Most of the 13 methods are based on NN. The exceptions are PORTER, SCRATCH, SSPro4 (based on bidirectional recurrent NN), SAM-T99sec (based on HMM) and Yaspin (based both on NN and HMM).

2.4.2 Secondary Structure Prediction Methods

In this section, we describe in detail two of the most famous secondary structure prediction methods: PHD[6] and PSIpred.[7] Both methods are based on NN and share similar network topology. The main difference between the two methods is the way evolutionary information is extracted from MSA and encoded into the NN input. Early version of PHD used HSSP pre-computed multiple alignments generated by MAXHOM. PSIpred uses the position-specific scoring matrix (PSSM) internally computed by PSI-BLAST. As discussed in [41], the improvement of PSIpred with respect to PHD is mostly due to the better alignments used to fed the NN. The better quality of the alignments is in part due to the growth of the databases and the filtering strategy used by Jones to avoid pollution of the profile through unrelated proteins. A more recent version of PHD uses PSSM input and it is called PHDpsi to distinguish it from the older implementation. The only difference between PHD and PHDpsi is the use of PSSM input instead of frequency profile input.

Also for all the other secondary structure predictors, the main source of information is the sequence profile or the PSSM. The main difference between the different approaches relies on the technique used to extract knowledge from these two sources of information. The particular technique is specific to the machine learning method used. Here we decided to describe only PHD and PSIpred because, historically, they represent the two most important step-forward in secondary structure prediction.

2.4.2.1 PHD

PHD has been described in [42]. The PHD method processes the input information in two different levels, corresponding to two different neural networks: (1) *sequence-to-structure NN* and (2) *structure-to-structure NN*. The final prediction is obtained by filtering the solution obtained from consensus between differently trained neural networks (3).

1. At the first level, the input units of the NN encode local information taken from sequence profiles (from PSSM in PHDpsi). For each residue position i, the local information is extracted from a window of 13 adjacent residues centered in i. For each residue position in the window, 22 input units are used: 20 units encode the corresponding column in the sequence profile, 1 unit is used to detect

[6] http://www.predictprotein.org/

[7] http://bioinf.cs.ucl.ac.uk/psipred/

when the position is outside the N/C-terminal region (1 if outside and 0 if not) and 1 unit accounts for the conservation weight at that position (see below for definition). The output of the first level NN consists of three nodes, one for each possible secondary structure element helix/strand/coil, corresponding to the state of the central residue in the window. The first level NN classifies (13-residues long) protein segments according to the secondary structure class of their central residue. This classification does not reflect the fact that different segments can be correlated, being, for example, consecutive and overlapping in the protein sequence. Particularly, at this level, the NN has no knowledge of the correlation between secondary structure elements. For example, it has no way to know that a helix consists of at least three consecutive elements.

2. The second level is introduced to take into account the correlation between consecutive secondary structure elements. The input of the second level NN is compiled from the output of the first level NN. For every residue position, the input unit encodes a window of 17 consecutive elements taken from the secondary structure prediction of the first NN. Every position in the window is encoded with 5 units: three for the predicted secondary structure, one to detect whether the position is outside the boundaries of the protein and one for the conservation weight. The output is set as in the first NN and, also in this case, corresponds to the state of the central residue in the window.
3. The consensus is a simple arithmetic average over (typically four) differently trained networks. The highest value of the three output units is taken as the final prediction. To every such prediction, a reliability index can be associated with the following formula
$$RI = \lceil 10 \cdot (o_1 - o_2) \rceil, \tag{2.4}$$
where o_1 and o_2 are the highest and the second highest values in the output vector, respectively. The prediction obtained is finally filtered (with the help of the reliability index) in order to fix some eventually unrealistic local predictions that neither the second level NN nor the consensus were able to detect (particularly, too short alpha-helix segments).

The conservation weight provides a score for positions in the MSA with respect to their level of conservation: the more conserved is a position the higher is the conservation weight score. Such a weight is contained in the HSSP database and it is defined by

$$CW_i = \frac{\sum_{r,s=1}^{N} w_{rs} \cdot \mathrm{sim}_{rs}^{i}}{\sum_{r,s=1}^{N} w_{rs}} \tag{2.5}$$

with

$$w_{rs} = 1 - \frac{1}{100} \cdot \mathrm{ident}_{rs},$$

where N is the number of sequences in the multiple alignment, ident_{rs} is the percentage of sequence identity (over the entire length) of sequences r, s and sim_{rs}^{i} is the value of the similarity between sequences r, s at position i according to the Dayhoff similarity matrix [8].

2.4.2.2 PSIpred

PSIpred has been described in [21]. The original implementation is based on neural networks. An almost equivalent implementation with SVM has been described in [48] and compared with the original version.

The neural network topology of PSIpred is very similar to the one used in PHD: in both methods the input is processed in two different levels, and the final result is obtained as the consensus between differently trained networks. The main differences are the lengths of the windows used in the first and second levels: in both networks PSIpred uses 15-residue long windows, while PHD uses lengths 13 and 17, respectively. Moreover, the conservation weight is not included in the input of PSIpred (it showed poor improvement also in PHD [42]). The most important difference between early PHD version and PSIpred is the way evolutionary information is treated. In particular, the position-specific scoring matrix (PSSM) is used to fed the NN instead of the classical frequency profile computed from MSA.

Here we review in detail the procedure used by Jones to produce meaningful position-specific profiles with PSI-BLAST, as described in [21]. Although PSI-BLAST is much more sensitive than BLAST in picking up distant evolutionary relationships, it must be used carefully in order to avoid false-positive matches. In particular, PSI-BLAST is very prone to incorporate repetitive sequences into the intermediate profiles. When this happens, the searching process tends to find highly scored matches with completely random sequences. In order to maximise the performances of PSI-BLAST, Jones builds a custom sequence data bank by first compiling a large set of non-redundant protein sequences and then by filtering the databank in order to remove low complexity regions [49], transmembrane segments [22] and regions which are likely to form coiled-coil regions (these filtering are now automatically performed by PSI-BLAST).

Finally, the input of the NN is computed from the PSSM of PSI-BLAST after three iterations, scaled to values between 0 and 1 with the logistic function $1/(1 + e^x)$, where x is the raw profile value.

2.5 Residue–Residue Contact Prediction

Residue–residue contact prediction refers to the prediction of the probability that two residues in a protein structure are spatially close to each other. Inter-residue contacts provide much information about the protein structure. A contact between two residues that are distant in the protein sequence can be seen as a strong constraint on the protein fold. If we could predict with high precision even a small set of (non-trivial) residue pairs in contact, we could use this information as extra constraints to guide the protein structure prediction. The prediction of inter-residue contact is a difficult problem, and no satisfactory improvements have been achieved in the last 10 years of investigation. On the other end, even if residue contact predictors are highly inaccurate, they still have higher accuracy compared to contact predictions derived from the best 3D structure prediction methods [45].

In the following sections, we describe the standards adopted for contact definition and contact prediction evaluation (Sect. 2.5.1). We next describe the most important statistics used to extract contact information from MSA (Sect. 2.5.2) and the best performing contact predictors, as evaluated in the last five CASP editions (Sect. 2.5.3).

2.5.1 EVAcon: Evaluation of Inter-Residue Contact Prediction Servers

Equivalently to EVAsec, the objectives of EVAcon[8] are to provide a continuous, fully automated and statistically significant analysis of inter-residue contact prediction servers. Differently from EVAsec, the statistics of EVAcon are not so frequently updated and only very few servers are monitored at the moment. Anyway, EVAcon provides the standards for contact definition and evaluation criteria for contact prediction. These measures are also those adopted at CASP meetings.

There are several ways to define inter-residue contacts; all definitions are more or less equivalent. In EVAsec, two residues are defined to be in contact if the Euclidean distance between the coordinates of their *beta carbon atoms* (C_β) is $\leq$8 Angstroms (Å) (Fig. 2.3). For Glycines, the coordinate of the *alpha carbon atom* (C_α) is considered instead of the C_β coordinate, which is missing (i.e., Glycines have only a unique carbon atom).

The most important measure for the evaluation of contact predictors is the accuracy of prediction. The accuracy of prediction is defined as

$$\frac{\text{Number of correctly predicted contacts}}{\text{Number of predicted contacts}}$$

Since contact predictors usually return the probability that two residues are in contact, the above formula is computed in slightly different way: the list of residue pairs is sorted in decreasing order according to the predicted contact probability, and the accuracy is computed by taking the first 2L, L, L/2, L/5 and L/10 (most probable) pairs, where L here denotes the length of the protein. More formally, the accuracy of prediction with respect to length $l \in \{$ 2L, L, L/2, L/5,L/10$\}$ is defined as

$$Acc_l = \frac{nc_l}{l}, \tag{2.6}$$

where nc_l is the number of correctly predicted contacts among the first l high-scored pairs. It makes sense to distinguish between short-range contacts (i.e., contacts between residues that are close in the protein sequence) and long-range contacts (i.e., contacts between residues that are distant in the sequence). Long-range contacts are much more sparse than short-range contacts, but they provide much more

[8] http://cubic.bioc.columbia.edu/eva/doc/intro_con.html

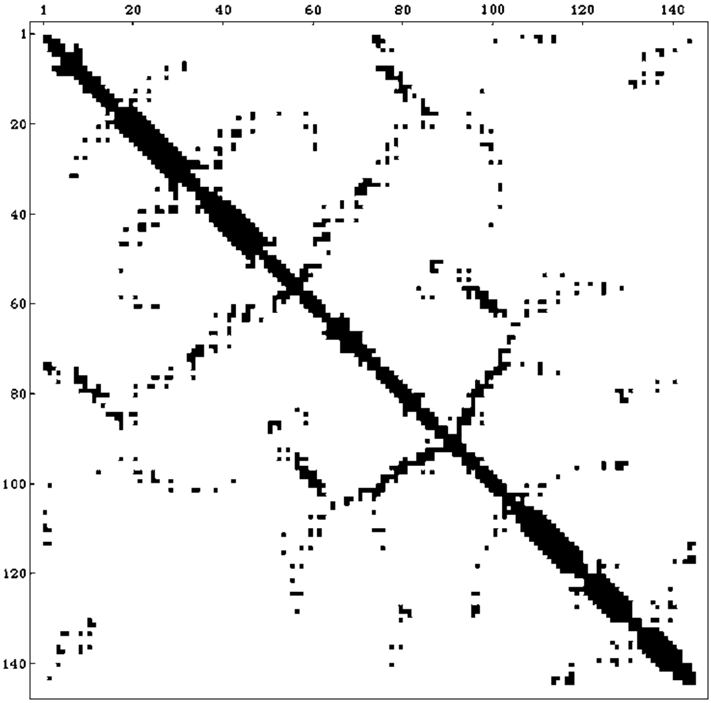


Fig. 2.3 Map of the C_β–C_β contacts at threshold 8 Å of the protein 1a3aA of Fig. 2.2. *Black dots* represent contacts

information about the protein structure and for this they are much more difficult to predict correctly. For this reason, for the calculation of the accuracy with (2.6), the predicted pairs of residues are split in three sets according to the separation of the two residues in the pair (i.e., the number of residues between them): short-range (from 6 to 11), medium-range (from 12 to 23) and long-range ($\geq$24) sequence separation. Residue contacts whose sequence separation is below 6 do not provide useful information about the protein folding and are not evaluated. Among all these measures, the most important evaluation parameter is the accuracy for sequence separation $\geq$24 and L/5 number of pairs.

Since the performances of contact predictors are not monitored as extensively as secondary prediction servers, the statistics about the state-of-the-art accuracy of inter-residue contact prediction is not very significant. According to the results obtained in the last two CASP events [11, 20], we can evaluate the state of the art in prediction accuracy as few percentage points above the 20% for sequence separation $\geq$24 and L/5 number of contacts.

2.5.2 Contact Prediction with Correlated Mutations

The most simple approach to predict residue–residue contacts in a protein is based on the evaluation of statistical properties derived from paired-columns of the MSA.

This approach relies on a very simple but significant hypothesis: since evolutionary mutations tend to preserve more protein structures than its sequence, residue substitutions must be compensated by other mutations in the spatially close neighbours in order to not destabilise the protein fold. That is, during the evolutionary course of protein sequences, a pair of residues in contact are more likely to co-mutate than residues not in contact. This basic idea has been exploited in the reverse direction: the probability of two residues to be in contact can be inferred by measuring how much changes in one column of the MSA (corresponding to one of the two residues) affect changes in the other column. There are various measures which can be used to extract correlated mutation statistics from the MSA. Despite the intensive investigation of correlated mutation-based methods, this approach alone resulted in a limited success in predicting residue–residue contacts. A possible explanation of these performances can be that the statistical measures exploited so far are too weak to discriminate between true correlation and background noise. Nevertheless, these methods are still interesting on their own, and they have been often used proficiently in conjunction with machine learning approaches for the contact prediction problem.

In the following sections, we shortly describe just few of the statistical measures used to evaluate correlated mutations; more detailed information can be found in [14, 19].

2.5.2.1 Pearson Correlation

The best known implementation of the correlated mutation approach [17] uses the Pearson correlation coefficients to quantify the amount of co-evolution between pair of sites.

The Pearson product-moment correlation coefficient is a measure of the linear dependence between two variables X, Y, and it is defined as

$$C(X,Y) = \frac{1}{N}\sum_{k=1}^{N}\frac{\left(X_k - \overline{X}\right)\left(Y_k - \overline{Y}\right)}{\sigma_X \sigma_Y}, \tag{2.7}$$

where N is the number of elements contained in X and Y, $\overline{X}$ is the average of X and σ_X is its standard deviation. The coefficient $-1 \leq C(X,Y) \leq 1$ quantifies the degree of linear dependence between X and Y. If $C(X,Y) = 1$, then X is equal to Y up to a linear transformation. In general, if $C(X,Y) \sim 1$, X and Y are considered positively correlated, not correlated if $C(X,Y) \sim 0$ and anti-correlated if $C(X,Y) \sim -1$.

To evaluate the Pearson correlation of a pair of sites (columns) i, j in an MSA, we have to define two substitution score vectors. Assuming that the MSA matrix M contains t aligned sequences, the substitution vector corresponding to position i is defined as

$$\begin{aligned} X = S_i = (&\delta(M_{1i}, M_{2i}), \delta(M_{1i}, M_{3i})\ldots, \\ &\delta(M_{1i}, M_{ti}), \delta(M_{2i}, M_{3i}), \ldots, \delta(M_{t-1i}, M_{ti})), \end{aligned}$$

where $\delta(M_{ki}, M_{li})$ is the score assigned to the substitution (mutation) $M_{ki} \rightarrow M_{li}$. The substitution vector $Y = S_j$ corresponding to position j is computed in the same way. The substitutions with gaps are not considered; hence, if $M_{ki} \rightarrow M_{li}$ is a gap-substitution then it is excluded from S_i and $M_{kj} \rightarrow M_{lj}$ is also excluded form S_j (the conversely holds for position j). The coefficient $C(S_i, S_j)$ quantifies the degree of linear correlation for the evolutionary mutations as observed at the ith column of the MSA with respect to the mutations occurring at the jth column. Perfectly conserved columns and columns with more than 10% of gaps are usually excluded from the analysis since they are uninformative.

This approach requires a similarity matrix to weight residue substitutions $M_{ki} \rightarrow M_{li}$: that is, the substitution vector is defined in terms of a scoring scheme $\delta(M_{li}, M_{ki})$. The substitution scores are generally provided by the McLachlan similarity matrix [28], which defines residue similarity in terms of their physico-chemical properties. The choice to use the McLachlan is not critical since there are several different similarity matrices that perform equally well [9]. Other related implementations of this method have been proposed (a comprehensive review can be found in [35]). These approaches differ from the original method [17] essentially in the measures adopted to weight the co-evolving substitutions.

2.5.2.2 Mutual Information

The mutual information measures the mutual dependence of two variables X, Y. It is defined as

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p_1(x) p_2(y)}, \tag{2.8}$$

where $p_1(x)$ is the marginal probability distribution of x in X, $p_2(y)$ is the marginal probability distribution of y in Y and $p(x,y)$ is the joint probability of x, y, i.e., the probability that x and y occur in conjunction. The mutual information is $I(X,Y) = 0$ if and only if X and Y are independent.

To evaluate the mutual information of two columns i, j of the MSA, we have to compute the marginal probabilities of residues occurring in each respective column and their joint probability. The variable X contains the different residues occurring in the ith column of the MSA, and $p_1(x), x \in X$ is the probability of residue x of being in the ith column, i.e., $p_1(x)$ is the frequency of residue x in the ith column of the MSA. The marginal probabilities of column j are computed in the same way. The joint probability $p(x,y), x \in X, y \in Y$ is the frequency of the pair x, y in the columns i, j of the MSA. In order to compute the mutual information of two positions i, j, the MSA is filtered from sequences containing a gap in position i or j. Note that, if either position i or j are perfectly conserved in the MSA, their mutual information reduces to 0.

A comparison in terms of prediction accuracy between Pearson correlation and mutual information has been analysed in [14]. According to this analysis, mutual information shows poor performances in contact prediction. Nevertheless, a more deep analysis described in [45] shows that the significance of the observed mutual information results in a much more strong measure for correlated mutations.

2.5.2.3 Joint Entropy

The joint entropy is a measure of how much entropy (variation) is contained in two variables X, Y. It is defined as

$$J(X,Y) = -\sum_{x \in X, y \in Y} p(x,y) \log p(x,y), \tag{2.9}$$

where $p(x,y)$ is the joint probability of x, y.

To compute the joint entropy of a pair of columns in the MSA, X, Y and $p(x,y)$ are defined as in Sect. 2.5.2.2. Note that for perfectly conserved positions, the joint entropy reduces to 0. Highly conserved positions are more likely to correspond to residues buried in the core of the protein, which is the most stable portion of a protein structure and thus less subjected to evolutionary mutations. Most of the residue–residue contacts are, in fact, localised in the core of a protein structure. Therefore, residue pairs with lower joint entropy are more likely to be in contact than pairs with higher entropy. In this sense, joint entropy is complementary to Pearson correlation and mutual information, which cannot extract information from highly conserved columns of the MSA (recall that perfectly conserved position are excluded from the Pearson analysis and have mutual information equal to 0).

2.5.3 Contact Prediction with Neural Networks

We describe the best performing NN contact predictors CORNET,[9] PROFcon,[10] and SAM-T06con,[11] as evaluated in the last five editions of CASP experiments (from CASP4 in 2000 to CASP8 in 2008).

All best known implementations of NN contact predictors have some common similarities. First of all, due to the high variability of protein lengths, the NN input cannot be set in order to directly encode the overall protein sequence. For this reason, the NN input encodes specific information related to pair of residues and only coarse-grained global features of the protein are taken into account. This information is usually derived from the MSA and from structural/statistical properties of the protein. We can identify three different kinds of information used in NN input units:

- *Local information*, derived from the respective local environments of the two residues;
- *Global information*, derived from the overall protein structure and/or sequence;
- *Paired-residue statistics*, which include statistical properties derived from paired columns of the MSA.

[9] http://gpcr.biocomp.unibo.it/cgi/predictors/cornet/pred_cmapcgi.cgi

[10] http://cubic.bioc.columbia.edu/services/profcon/

[11] http://compbio.soe.ucsc.edu/SAM_T06/T06-query.html

All NN predictors described here differ essentially only by the features chosen to capture these three different kinds of information.

The output layer of the NN contains a unique node, which during the training phase is set to 1 if the two residues are in contact and to 0 if they are not. Accordingly, in the prediction phase the output of the NN (a value between 0 and 1) is interpreted as the probability that the two input residues are in contact.

In order to filter most of the (uninformative) data related to local contacts, the set of training examples is computed only from residue pairs whose sequence separation is larger than some threshold, typically >6. Moreover, to avoid the overestimation of non-contacts (which are much more abundant than contacts), the training examples are usually balanced. The balancing is generally obtained by randomly selecting only a fraction of the negative examples (typically 5%) from each epoch of the training phase. This technique has the effect from speeding up the learning process and assures that most of the negative examples are seen by the NN.

The performances and the limits of NN predictors are strictly related to their input encodings. Different from the secondary structure prediction problem, the contact probability is not a property that can be inferred locally, since it is a consequence of repulsive and attractive inter-atomic forces over all the protein sequence. Due to the limit imposed by the different protein lengths, the NN predictors are forced to infer global information about the protein structure mostly from local information only. This is probably the main reason why residue–residue contact prediction is not as successful as secondary structure prediction. Nevertheless, the NN-based approaches are actually the state of the art in residue–residue contact prediction and they provide much better performances than the contact prediction derived from tertiary structure modeling (for free-modeling domains).

In the following sections, for each NN predictor, we focus on the specific encoding of the input information. More detailed description of the implementations together with the analysis of their performances can be found elsewhere [13,38,45].

2.5.3.1 CORNET

The implementation of CORNET and its performances have been described in [12, 13].

In total, the input encodings requires 1,071 units. Most of the NN input encodes paired-residue statistics. For each residue pair i, j $(j > i + 6)$ in the protein sequence, the NN input encodes local information (a) in terms of sequence conservation of positions i, j and in terms of predicted secondary structure of their immediate neighbours, i.e., the two windows $[i - 1, i + 1]$ and $[j - 1, j + 1]$ are considered. Two distinct paired-residue statistics are used (b): Pearson correlated mutations and paired evolutionary information as observed in the two neighbouring windows. No global information is taken into account.

a. For each position in the two neighbouring windows three input units encode the secondary structure information (alpha/beta/coil). If the secondary structure in one position is predicted as alpha, then the corresponding entry in the input unit is 1 and the remaining two entries are set to 0. The same holds for the other secondary structure elements. When the neighbouring window is outside the boundaries of protein, all entries of the secondary structure input units are set to 0. The sequence variability, as computed in [15], is included only for positions i and j (2 units).

b. The evolutionary information for the pair i, j is encoded as an input vector containing $210 = 20 \cdot (20 + 1)/2$ elements, one entry for each distinct pair of aminoacids (symmetric pairs are considered equivalent). Every entry of the vector contains the frequency of the occurrence of the related pair of amino-acids in the multiple alignment with respect to positions i, j. The evolutionary information of the neighbours of i, j is also taken into account. The positions considered to introduce the evolutionary information are $(i - 1, j - 1), (i + 1, j + 1)$ (parallel pairings) and $(i - 1, j + 1), (i + 1, j - 1)$ (anti-parallel pairings). The correlated mutation information (1 unit) is defined as described in (2.7). For perfectly conserved positions, the correlation between i and j is set by default to 0 and for positions with more than 10% of gaps to -1.

2.5.3.2 PROFcon

The implementation of PROFcon and its performances have been described in [38].

In total, the input encodings require 738 units. For every pair of residues i, j, the neural network input incorporates local information from the neighbours of i, j and from their connecting segment (a). Several global properties of the protein are taken into account (b) but not the paired-residue statistics.

(a) The local information of the two residues is derived from two windows of width nine centered in i, j and from the segment connecting i, j. The connecting segment information is captured by taking a window of five consecutive residues from $k - 2$ to $k + 2$ where $k = \lceil i - j \rceil$. Each residue position in the three windows is described by the frequency of occurrence of the 20 amino acid types in that position (20 input units plus 1 more unit to detect when the position is outside the boundaries of the protein), predicted secondary structure (4 units, helix/strand/coil and reliability of the prediction at that position as defined in (2.4)), predicted solvent accessibility (3 units, buried/exposed and prediction reliability) and conservation weight (1 unit) as defined in (2.5). Some more features are introduced to better characterise the biophysical properties of the pair i, j (7 input units: hydrophobic-hydrophobic, polar-polar, charged-polar, opposite charges, same charges, aromatic-aromatic, other) and if they are in low-complesity regions, as computed by the SEG software (2 input units). Global features of the entire connecting segment are also considered: amino acid composition (20 units), secondary structure composition (4 units) and the fraction of SEG-low-complexity residues in the whole connecting segment

(1 node). Finally, the length of the segment connecting i and j is encoded in 11 input units corresponding to sequence separations 6, 7, 8, 9, 10–14, 15–19, 20–24, 25–29, 30–39, 40–49, >49.

(b) This global information includes amino acid composition of the entire protein (20 units), secondary structure composition (3 units) and protein length (4 units, lengths 1–60, 61–120, 121–240 and >240).

2.5.3.3 SAM-T06con

This NN contact predictor is included in the protein structure prediction architecture SAM-T06. The implementation of the contact predictor and its performances have been described in [45].

In total, the input encoding of the NN requires 449 units. The local information (a) is accounted by taking a windows of length five centered in each one of the two residues. Four distinct paired-residue statistics are used (b) and just the length of the protein is taken into account as global information (c).

(a) For each position in the two windows, the NN input encodes the amino acids distribution according to a Dirichlet mixture regularizer [46] (20 units), the predicted secondary structure and predicted burial [25] (13 and 11 units, respectively). Moreover, the entropy of the amino acids distribution (1 unit for each window) and the logarithm of the sequence separation between the two residues (1 unit) are included.

(b) The NN input encodes four paired-residue statistics (1 input unit for three of them and 2 for the last one). The most simple statistics counts the number of different pairs observed in the MSA columns corresponding to the two residues. Other statistics considered are the joint entropy (2.9), the propensity of contact, and a mutual information-based statistics (2.8). For these three last measures, the logarithm of the rank of the statistic's value is taken into the input, except for the mutual information for which both the logarithm of the rank and the exact value are added. The rank of a statistic value is computed as the rank of the value in the list of values for all pairs of columns.

The propensity for two residue to be in contact is the log odds of a contact between the residues vs. the probability of the residues occurring independently. This measure has been slightly modified in order to give more weight to high-separation with respect to low-separation contacts. Here the mutual information statistics is introduced by computing its p-value (i.e., the probability of seeing the observed mutual information by chance). The significance of the mutual information shows better performances in contact prediction than the statistics itself, as computed in (2.8). More detailed information about the propensity of contact and the mutual information-based statistics can be found in [45].

(c) The only global information added is the logarithm of the length of the protein (1 unit).

2.6 Conclusions

In this chapter, we presented two different aspects of the protein structure prediction problem: the prediction of protein secondary structure, which is simpler in its formulation than the protein folding problem and from which sequential annotations can be derived, and the most demanding problem of residue contact prediction in proteins. The first relevant message from our analysis of the current state-of-the-art methods is that a key-role is played by evolutionary information. This knowledge, which can be exploited by using different multiple sequence alignment methods, is one of the major resources to identify relevant domains of the protein that are related to secondary structure elements or packing regions. A second relevant message is that the most successful predictors are based on machine-learning tools, indicating that for the described tasks (at least up-to-now) bottom-up approaches compete favorably with the methods that directly predict the 3D structure of the proteins.

References

1. Aloy, P., Stark, A., Hadley, C., Russell, R.B.: Predictions without templates: new folds, secondary structure, and contacts in CASP5. Proteins **53**, 436–456 (2003)
2. Altschul, S.F., Madden, T.L., Schffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**, 3389–3402 (1997)
3. Bartoli, L., Capriotti, E., Fariselli, P., Martelli, P.L., Casadio, R.: The pros and cons of predicting protein contact maps. Methods Mol Biol. **413**, 199–217 (2008)
4. Benner, S.A., Gerloff, D.: Patterns of divergence in homologous proteins as indicators of secondary and tertiary structure: a prediction of the structure of the catalytic domain of protein kinases. Adv. Enzyme Regul. **31**, 121–181 (1991)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2007)
6. Chou, P.Y., Fasman, G.D.: Conformational parameters for amino acids in helical, beta-sheet, and random coil regions calculated from proteins. Biochemistry **13**, 211–222 (1974)
7. Cozzetto, D., Tramontano, A.: Advances and pitfalls in protein structure prediction. Curr Protein Pept Sci. **9**, 567–577 (2008)
8. Dayhoff, M.O.: Atlas of Protein Sequence and Structure. National Biomedical Research Foundation, Washington DC (1978)
9. Di Lena, P., Fariselli, P., Margara, L., Vassura, M., Casadio, R.: On the Upper Bound of the Prediction Accuracy of Residue Contacts in Proteins with Correlated Mutations: The Case Study of the Similarity Matrices. Lecture Notes in Computer Science 5488, 210–221 (2009)
10. Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. **32**, 1792–1797 (2004)
11. Ezkurdia, I., Graña, O., Izarzugaza, J.M., Tress, M.L.: Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8. Proteins **77**, 196–209 (2009)
12. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Prediction of contact maps with neural networks and correlated mutations. Protein Eng. **14**, 835–843 (2001)
13. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. Proteins **5**, 157–162 (2001)
14. Fodor, A.A., Aldrich, R.W.: Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. Proteins **56**, 211–221 (2004)

15. Garcia-Boronat, M., Diez-Rivero, C.M., Reinherz, E.L., Reche, P.A.: PVS: a web server for protein sequence variability analysis tuned to facilitate conserved epitope discovery. Nucleic Acids Res. **36**, 35–41 (2008)
16. Garnier, J., Osguthorpe, D.J., Robson, B.: Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. J. Mol. Biol. **120**, 97–120 (1978)
17. Göbel, U., Sander, C., Schneider, R., Valencia, A.: Correlated mutations and residue contacts in proteins. Proteins **18**, 309–317 (1994)
18. Graña, O., Baker, D., MacCallum, R.M., Meiler, J., Punta, M., Rost, B., Tress, M.L., Valencia, A.: CASP6 assessment of contact prediction. Proteins **61**, 214–224 (2005)
19. Horner, D.S., Pirovano, W., Pesole, G.: Correlated substitution analysis and the prediction of amino acid structural contacts. Brief. Bioinform. **9**, 46–56 (2008)
20. Izarzugaza, J.M., Graña, O., Tress, M.L., Valencia, A., Clarke, N.D.: Assessment of intramolecular contact predictions for CASP7. Proteins **69**, 152–158 (2007)
21. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. **292**, 195–202 (1999)
22. Jones, D.T., Taylor, W.R., Thornton, J.M.: A model recognition approach to the pre-diction of all-helical membrane protein structure and topology. Biochemistry **33**, 3038–3049 (1994)
23. Kabsch, W., Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers **22**, 2577–2637 (1983)
24. Karplus, K., Barrett, C., Hughey, R.: Hidden Markov models for detecting remote protein homologies. Bioinformatics **14**, 846–856 (1998)
25. Karplus, K., Katzman, S., Shackleford, G., Koeva, M., Draper, J., Barnes, B., Soriano, M., Hughey, R.: SAM-T04: what is new in protein-structure prediction for CASP6. Proteins **61**, 135–142 (2005)
26. Lesk, A.: Introduction to Bioinformatics. Oxford University Press, London (2006)
27. Lin, K., Simossis, V.A., Taylor, W.R., Heringa, J.: A simple and fast secondary structure prediction method using hidden neural networks. Bioinformatics **21**, 152–159 (2005)
28. McLachlan, A.D.: Tests for comparing related amino-acid sequences. Cytochrome c and cytochrome c 551. J. Mol. Biol. **61**, 409–424 (1971)
29. Notredame, C., Higgins, D.G., Heringa, J.: T-Coffee: A novel method for fast and accurate multiple sequence alignment. J Mol Biol. **302**, 205–217 (2000)
30. Ouali, M., King, R.D.: Cascaded multiple classifiers for secondary structure pre-diction. Protein Sci. **9**, 1162–1176 (2000)
31. Pauling, L., Corey, R.B.: Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets. Proc. Natl. Acad. Sci. USA **37**, 729–740 (1951)
32. Pauling, L., Corey, R.B., Branson, H.R.: The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. Proc. Natl. Acad. Sci. USA **37**, 205–211 (1951)
33. Pollastri, G., McLysaght, A.: Porter: a new, accurate server for protein secondary structure prediction. Bioinformatics **21**, 1719–1720 (2005)
34. Pollastri, G., Przybylski. D., Rost, B., Baldi, P.: Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins **47**, 228–235 (2002)
35. Pollock, D.D., Taylor, W.R.: Effectiveness of correlation analysis in identifying protein residues undergoing correlated evolution. Protein **10**, 647–657 (1997)
36. Porollo, A., Adamczak, R., Wagner, M., Meller, J.: Maximum Feasibility Approach for Consensus Classifiers: Applications to Protein Structure Prediction. In proceedings of CIRAS 2003
37. Przybylski, D., Rost, B.: Alignments grow, secondary structure prediction improves. Proteins **46**, 197–205 (2002)
38. Punta, M., Rost, B.: PROFcon: novel prediction of long-range contacts. Bioinformatics **21**, 2960–2968 (2005)
39. Raghava, G.P.S.: APSSP2: A combination method for protein secondary structure prediction based on neural network and example based learning. CASP5 A-132 (2002)
40. Rost, B.: http://cubic.bioc.columbia.edu/predictprotein

41. Rost, B.: Rising accuracy of protein secondary structure prediction. In: Chasman D (ed.) Protein structure determination, analysis, and modeling for drug discovery, pp. 207–249. Dekker, New York (2003)
42. Rost, B., Sander, C.: Prediction of protein secondary structure at better than 70% accuracy. J. Mol. Biol. **232**, 584–599 (1993)
43. Rost, B., Sander, C.: Third generation prediction of secondary structures. Methods Mol. Biol. **143**, 71–95 (2000)
44. Sander, C., Schneider, R.: Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins **9**, 56–68 (1991)
45. Shackelford, G., Karplus, K.: Contact prediction using mutual information and neural nets. Proteins **69**,159–164 (2007)
46. Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I.S., Haussler, D.: Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. Comput. Appl. Biosci. **12**, 327–345 (1996)
47. Thompson, J.D., Higgins, D.G., Gibson, T.J.: 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 (1994)
48. Ward, J.J., McGuffin, L.J., Buxton, B.F., Jones, D.T.: Secondary structure prediction with support vector machines. Bioinformatics **19**, 1650–1655 (2003)
49. Wootton, J.C., Federhen, S.: Statistics of local complexity in amino acid sequences and sequence databases. Comput. Chem. **17**,149–163 (1996)

Chapter 3
Secondary Structure Classification of Isoform Protein Markers in Oncology

Gregorio Patrizi, Claudio Cifarelli, Valentina Losacco, and Giacomo Patrizi

Abstract The determination of the secondary structure of proteins can be considered a relevant procedure to characterise isoform protein markers for cancer and other pathologies. Their recognition is modeled as a classification problem and solved using a nonlinear complementarity problem restricted to binary variables. The procedure will be tested on an available data sets of proteins to determine the differences in the isoforms which affect the pathologies. Recognition accuracy is attributable to the implementation of a nonlinear binary optimization problem and a strict statistical methodology that does not require extraneous assumptions.

3.1 Introduction

Proteins assume a well-defined three-dimensional structure, which is determined principally by their amino acid sequence. Per contra the structure of a protein determines its function and the sequence of amino acids which compose a given protein determines its spatial relationship. Also in general, isoforms are different forms of a protein that may be produced from small differences between alleles of a gene or from the same gene by alternating splicing which differs in structure, but may also be due to differences in their spatial characteristics, arising from small differences in the amino acid chain, or other mechanisms not well defined [6,20,28]. There are 20 different amino acids which form proteins, and their combinations may be presented in four levels of structures: the primary structure characterized by the sequence of amino acids, a secondary structure in which their structure is distinguished by the folding characteristics of the protein, classified by segments of the amino acids indicated by α-helix, a helical type structure, β-strand, a sheet-like structure, and a residual class indicated as coil. Often the classification of segments is extended to eight to ten classes, depending on requirements. The tertiary structure consists of

G. Patrizi (✉)
Dipartimento di Scienze Chirurgiche, “Sapienza” Universita di Roma, Rome, Italy
e-mail: g.patrizi@caspur.it; g_patrizi@yahoo.com

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_3,

the triple angles between successive amino acids, and in the quaternary structure, proteins are lumped together in bigger composites [7].

Appropriate methods that achieve high sensitivity and specificity levels are required for early diagnosis of cancer for patient survival, and successful diagnosis and prognosis of the disease are crucial. Protein markers are proteins usually present in the blood, and their isoform incidence may vary with oncoming oncological malignancies, so the analysis of biomarkers in blood and other body fluids is applied in the detection of the disease by applying various methods of molecular recognition [29] to determine the relative proportions of different isoforms in the blood [24]. Suitable procedures must be defined to recognise and obtain samples of the various isoforms, which can be determined, principally, by electrophoresis, probes, or protein microarrays [1, 34, 37].

The relationships that bind consecutive amino acids in a protein may be characterised by suitable class labels indicating the structural relationship between these amino acids. Using analytical mathematical classification methods, on the basis of the primary structure of a protein, the secondary structure is predicted and so the particular isoform is consequently identified. Classification consists of assigning an entity to a class, in such a way that similar entities are assigned to the same class. The concept of similarity may be definable a priori and in this case a problem in taxonomy is formulated [13, 26], or the class and the membership status of the given entities must be identified on the basis of some previously inferential set of characteristics of the objects, a process basic to human knowledge [33], and the classification results that are obtainable will depend on the information structure provided [21]. For the algorithm that will be described, the average precision of 87% is obtained [5]. At this level of precision, isoform differences of proteins are detectable; so the approach suggested is novel.

The aim of this chapter is to present a nonlinear complementarity algorithm, limited to binary variables, to implement a classification algorithm to determine the secondary structure of isoforms of proteins which constitute markers in oncology through their secondary structure with a high precision. Thus, the outline of this chapter is as follows. In the next section, the properties of the structure and the dynamics of isoforms will be examined, in particular, with regard to their folding characteristics. In Sect. 3.3, the classification algorithm is presented, while in the following section the coherent statistical properties of the algorithm will be derived. In Sect. 3.5, some experimental results regarding isoforms will be presented. Finally in Sect. 3.6, the relevant conclusions will be drawn.

3.2 Structural Diversity of Isoform

Many biosensors for cancer are being studied and experiments are undertaken [29]. The identification of a group of proteins consistently changing in relation to the disease is important in cancer research [29] and raises deep problems in determining the detailed secondary and tertiary structure of these proteins. For instance the identification of a protein by determining its primary structure or by empirical experimental methods may not be sufficient; so a more complete functional analysis of the isoform of a protein should be assessed by exploiting methods to determine accurate secondary or tertiary structure. For example, tumor-screening markers for prostate cancer consist in the identification of acid phosphate; however, a large number of false-negative results limit the usefulness of this marker [27]. The determination of a set of markers by identifying its secondary structure could result in higher levels of specificity and sensitivity for the different types of markers. Again for patients showing serum PSA levels between 4 and 10 mg ml^{-1}, the positive predictive value is only 18–25% (mean 21%) and the diagnostic precision might be increased by distinguishing more finely the markers, through determining more precise structural characteristics [27].

The recognition in situ carcinomas and pre-invasive foci of the primary breast epithelial tumors, hS100A7 expression often decrease in invading tumor foci; however, its persistent expression in invasive carcinomas is associated with poor prognosis [34]. The hS100A7 (psoriasin) protein belongs to a large multigenic family of calcium-binding EF-hand S100 proteins [25]. As evolutionary late genes, hS100A7 and hS100A15 are highly similar paralogs (93% sequence identity), most similar among all S100 gene family. These proteins can be distinguished through the application of specific antibodies to unique peptide sequences in the amino terminus. The high homology of these proteins makes them often difficult to distinguish when coexpressed, so their distinct biological functions compel an analysis of their dual presence and potential contribution to normal breast and to cancer pathologies, while both proteins contribute unique functional elements for breast physiology and tumorigenesis, so the secondary or higher structures could distinguish directly their functional characteristics.

The taxonomic characterization of proteins and the similarity of isoforms with respect to disease must be analyzed formally to enact suitable prognosis procedures. For instance bone and soft tissue sarcomas may be treated in some cases, but it is held that improvements may be expected through global investigations of the molecular backgrounds associated with the clinicopathologic characteristics of tumors [15]. Moreover, a human gene (TP73) is a part of the p53 family of transcription factors, which may form multiple protein isoforms. Careful studies of the functions and the structure of the data available suggest to enrich the characterisation of the amino-terminally truncated p73 isoforms because of their role in driving cellular responses to anticancer agents and tumor growth control [4].

3.3 The Classification Algorithm

A set of objects may be specified by a set of common *attributes*, which are then assigned to certain *classes*. The *classification* problem consists in determining a mapping from the set of objects, characterised by the set of common attributes, to the set of classes. To define such a mapping, a *training set* is required with a set of objects classified in known classes [18].

Definition 3.1. A subset of a dataset is termed a training set if every entity in a given set has been assigned a class label.

Definition 3.2. Suppose there is a set of entities E and a set $P = \{P_1, P_2, \ldots, P_n\}$ of subsets of the set of entities, i.e. $P_j \subseteq E, j \in J = \{1, 2, \ldots, n\}$. A subset $\hat{J} \subseteq J$ forms a cover of E if $\bigcup_{j \in \hat{J}} P_j = E$. If, in addition, for every $k, j \in \hat{J}, j \neq k$, $P_j \cap P_k = \emptyset$ it is a partition.

Definition 3.3. The dataset is coherent if there exists a partition, which satisfies the following properties:

1. The relations defined on the training set and in particular the membership classes, defined over the data set, consist of disjoint unions of the subsets of the partition.
2. Stability: The partition is invariant to additions to the dataset. This invariance should apply both to the addition of duplicate entities and to the addition of new entities obtained in the same way as the objects under consideration.
3. Extendibility: If the dimension of the attributes of the set considered is p and it is augmented, so that the basis will be composed of $p + 1$ attributes, then the partition obtained by considering the smaller set will remain valid even for the extension, as long as this extension does not alter the relations defined on the dataset. Thus, the labels characterizing the training set are correct under either dimensional space.

Such a dataset is experimentally stable and precise partitions of the dataset can be obtained.

Definition 3.4. A dataset is linearly separable if there exist linear functions such that the entities belonging to one class can be separated from the entities belonging to the other classes. It is pairwise linearly separable, if every pair of classes is linearly separable. A set is piecewise separable if every element of each class is separable from all the other elements of all the other classes.

Clearly if a set is linearly separable, it is pairwise linearly separable and piecewise separable, but the converse is not true.

Theorem 3.1. *If a dataset is coherent then it is piecewise separable.*

Proof. By the Definition 3.3, a partition exists for a coherent dataset and therefore there exists subsets $P_j \subseteq E, j \in J = \{1, 2, \ldots, n\}$ such that for every $j \neq k \in J$, $P_j \cap P_k = \emptyset$, as indicated by Definition 3.2.

A given class is formed from distinct subsets of the partition, so no pattern can belong to two classes. Therefore, each pattern of a given class will be separable from every pattern in the other subsets of the partition. Consequently, the dataset is piecewise separable.

Theorem 3.2. *Given a dataset which does not contain two identical patterns assigned to different classes, then a correct classifier can be formulated which realizes the given partition on this training set.*

Proof. The proof is trivial, since if the dataset does not contain two identical patterns that belong to different classes, each pattern or group of identical patterns can be assigned to different subsets of the partition. This classifier is necessarily correct and on this basis subsets can be aggregated, as long as the aggregated subsets of different classes remain disjoint.

Corollary 3.1. *Given that the training set does not contain two or more identical patterns assigned to different classes, the given partition yields a completely correct classification of the patterns.*

Theorem 3.1 and the distinction introduced in the corollary are relevant to characterize the dataset and the training set to ensure the avoidance of the juxtaposition property, i.e. two identical patterns belong to different classes, entails that the Bayes error is zero [8].

The classification algorithm to be formulated may be specified as a combinatorial problem in binary variables [18, 21]. Consider a training set with n patterns represented by appropriate feature vectors indicated by $x_i \in \mathbb{R}^p, \forall i = 1, 2, \ldots, n$ and grouped in c classes. An upper bound is selected for the number of barycenters that may result from the classification, which can be taken “ad abundantiam” as m, or on the basis of a preliminary run of some classification algorithm. Hence, the initial barycenter matrix will be a $p \times mc$ matrix which is set to zero. The barycenters when calculated will be written in the matrix by class. Thus, a barycenter of class k will occupy a column of the matrix between $(m(k-1)+1)$ and mk. The feature vectors can be ordered by increasing class label. Thus, the first n_1 columns of the training set matrix consists of patterns of class 1, from $n_1 + 1$ to n_2 of class 2 and in general from $n_{k-1} + 1$ to n_k of class k.

Let:

- $x_i \in \mathbb{R}^p$: The p dimensional pattern vector of pattern i,
- c classes are considered, $k = 0, 1, \ldots, (c-1)$. Let the number of patterns in class c_k be indicated by n_k, then the n patterns can be subdivided by class so that $n = \sum_{k=0}^{c-1} n_k$,
- $z_j \in \{0, 1\}$ $\{j = 1, 2, ..mc\}$ if $z_j = 1$, then the barycenter vector $j \in \{mk + 1\}$ $, \ldots, m(k+1)\}$ belonging to recognition class $c_k \in \{0, \ldots, c-1\}$,
- $y_{ij} \in \{0, 1\}$, which indicates that the pattern i has been assigned to the barycenter j $(y_{ij} = 1)$,
- $t_j \in \mathbb{R}^p$ so the sum of the elements of the vectors of the patterns assigned to barycenter $j = \{1, 2, \ldots, mc\}$,
- M a large scalar.

Consider the following optimization problem defined in these variables:

$$\text{Min } Z = \sum_{j=1}^{mc} z_j \tag{3.1}$$

$$\text{s.t.} \sum_{j=km+1}^{m(k+1)} y_{ij} - 1 \geq 0 \qquad \forall k = 0, 1, \ldots, (c-1); \forall i = n_{k-1}+1, \ldots, n_k \tag{3.2}$$

$$-\sum_{i=1}^{n}\sum_{j=1}^{mc} y_{ij} + n \geq 0 \tag{3.3}$$

$$Mz_j - \sum_{i=1}^{n} y_{ij} \geq 0 \qquad \forall j = 1, 2, \ldots, mc \tag{3.4}$$

$$t_j - \sum_{i=1}^{n} x_i y_{ij} \geq 0 \qquad \forall j = 0, 1, \ldots, mc \tag{3.5}$$

$$-\sum_{j=1}^{mc}\left(t_j - \sum_{i=1}^{n} x_i y_{ij}\right) \geq 0 \tag{3.6}$$

$$\begin{aligned}&\left(x_i - \frac{t_h}{\sum_{s=lm+1}^{m(l+1)} y_{sh}}\right)^T \left(x_i - \frac{t_h}{\sum_{s=lm+1}^{m(l+1)} y_{sh}}\right)\\ &- \sum_{j=km+1}^{m(k+1)} \left(x_i - \frac{t_j}{\sum_{r=km+1}^{m(k+1)} y_{rj}}\right)^T \left(x_i - \frac{t_j}{\sum_{r=km+1}^{m(k+1)} y_{rj}}\right) \times y_{ij} \geq 0\\ &\forall i = 1, 2, \ldots, n; \quad h = 1, 2, \ldots, mc; \quad k, l = 0, 1, \ldots, c-1;\end{aligned} \tag{3.7}$$

$$z_j, y_{ij} \in \{0, 1\} \tag{3.8}$$

The nonlinear optimization problem (3.1)–(3.8) in binary values will solve the classification problem for the problem. The nonlinear complementarity problem in binary variables will be solved through successive linear complementarity problems in binary variables, using a linear programming technique with parametric variation in one scalar variable [22] which has given good results [9].

The solution of this optimization problem assigns each pattern to a mean vector, called a barycenter ($z_j, j = 1, 2, \ldots, mc$), whose values are given by the vectors $t_j \in \mathbb{R}^p$, $j = \{1, 2, \ldots, mc\}$ divided by the number of patterns assigned to that barycenter. The least number of barycenters (3.1) which will satisfy the stated constraints is determined.

The n constraints (3.2) and (3.3) state that each feature vector from a pattern in given class must be assigned to some barycenter vector of that class. As patterns and barycenters have been ordered by class, the summation should be run over the appropriate index sets.

The mc constraints (3.4) impose that no pattern be assigned to a nonexisting barycenter.

The constraints (3.5) and (3.6) determine the vector of the total sum element by element assigned to a barycenter, while for the set of inequalities (3.7) indicate that each feature vector must be nearer to the assigned barycenter of its own class than to any other barycenter. Should the barycenter be null, this is immediately verified, while if it is non zero, this must be imposed. The inequality (3.8) indicates that the vectors $z \in \mathbb{R}^{mc}$ and $y \in \mathbb{R}^{nmc}$ are binary.

The solution will determine that each pattern of the training set is nearer to a barycenter of its own class than to a barycenter of another class. Each barycenter has the class label of the patterns assigned to it, which will belong by construction to a single class. This defines a partition of the pattern space.

A new pattern can be assigned to a class by determining its distance from each barycenter formed by the algorithm and then assigning the pattern to the class of the barycenter to which it is nearest.

In addition, the optimization problem (3.1)–(3.8) may be formulated as a nonlinear complementarity problem facilitating the proof of convergence and termination of the algorithm. The nonlinear complementarity formulation is a statement of the Karush–Kuhn–Tucker condition of an optimization problem [19], and therefore, one of the solutions of the nonlinear complementarity problem will be a solution to that optimization problem

To demonstrate that the algorithm will converge to an optimal solution, consider the domain of the optimization problem to be over $\mathbb{R}^N$ a convex space.

$$F(w) \geq 0 \qquad F : \mathbb{R}^N \rightarrow \mathbb{R}^N \tag{3.9}$$

$$w \geq 0 \qquad w \in \mathbb{R}^N \tag{3.10}$$

$$w^T F(w) = 0, \tag{3.11}$$

where w comprises all the variables to be determined, the binary variables and the lagrangian multipliers of the inequalities.

This problem can be written as a variational inequality:

$$F(w)^T (u - w) \geq 0 \tag{3.12}$$

$$w \geq 0 \tag{3.13}$$

$$\forall \quad u \geq 0. \tag{3.14}$$

The solutions of the two problems are identical and the following results have been demonstrated [5].

Theorem 3.3. *Let $K \subset \mathbb{R}^N$ be a non empty, convex and compact set and let $F : K \rightarrow K$ be a continuous mapping. The following are equivalent:*

1. *There exists a fixed point $w^* \in K$ for this mapping,*
2. *The variational inequality (3.12) and (3.14) has a solution,*
3. *The nonlinear complementarity problem (3.9)–(3.11) has a solution*

Consider the nonlinear complementarity problem (3.9)–(3.11) and limit its solution to occur within a trust region set, defined by a set of linear inequalities which can be so indicated:

$$Dw \geq d, \tag{3.15}$$

such that this set defines a bounding polyhedron of appropriate dimension in the given space, which may be added to problem (3.1)–(3.8) which can be reformulated in the form of the system (3.9)–(3.11). Thus, consider the application $F : R^N \to R^N$ and expand it in a Taylor series around a point $w' \in R^N$ to get:

$$F(w) = F(w') + \nabla F(w')(w - w'), \tag{3.16}$$

then for any $\varepsilon > 0$, there exists a scalar $r > 0$ such that:

$$\left\| F(w) - F(w') + \nabla F(w')(w - w') \right\| \leq \varepsilon \left\| w - w' \right\|, \quad \forall \left\| w - w' \right\| \leq r. \tag{3.17}$$

Thus, in a small enough neighborhood, the approximation of the nonlinear complementarity problem (3.9)–(3.11) by a linear complementarity problem (LCP) will result sufficiently accurate; therefore, the following linear approximation can be solved iteratively:

$$Mw + q \geq 0 \tag{3.18}$$

$$w \geq 0 \tag{3.19}$$

$$w^T (Mw + q) = 0, \tag{3.20}$$

where M and q are appropriate linear approximations to the functional forms (3.9)–(3.11), and by construction, the subspace of the Eucledian space is bounded and closed; thus, the convergence of the algorithm can now be demonstrated as R^N in a convex space, so take a point $w' \in R^N$ such that $F(w') \geq 0$ and therefore feasible. Determine a neighbourhood, as large as possible, which can be indicated by:

$$Q = \left\{ w \mid \left\| w - w' \right\| \leq r \right\}, \tag{3.21}$$

where r is the coefficient defined above in (3.17).

Suppose that the acceptable tolerance to our solution is ε_5 so that if $(w^*)^T F(w^*) \leq \varepsilon_5$, then the solution is accepted. In this case, impose that:

$$\varepsilon r \leq \frac{\varepsilon_5}{\alpha}. \tag{3.22}$$

The local convergence of the algorithm is established in the following theorem.

Theorem 3.4. *If the linear complementarity problem has a solution w^* where all the trust region constraints are not binding, then such a solution is also a solution to the nonlinear complementarity problem (3.9)–(3.11) for which $F(w^*) \geq 0$ and $(w^*)^T F(w^*) \leq \varepsilon_5$.*

Proof. Consider the solution w^* of the linear complementarity problem (3.18)–(3.20). Recall that $\alpha \geq e^T w^*$ by construction and without loss of generality, take $\alpha > 1$. Consider this solution applied to the nonlinear complementarity problem, there will result:

$$\left\| F\left(w^{*}\right) - F\left(\hat{w}\right) + \nabla F\left(\hat{w}\right)\left(w^{*} - \hat{w}\right)\right\| \leq \varepsilon \left\| w^{*} - \hat{w}\right\| \leq \varepsilon r < \varepsilon_5 \tag{3.23}$$

For the complementarity condition

$$(w^{*})^{T} F(w^{*}) = (w^{*})^{T} \left(F(w^{*} - F(\hat{w}) + \nabla F(\hat{w})(w^{*} - \hat{w})\right) \leq \|w^{*}\| \, \varepsilon r \leq \varepsilon_5, \tag{3.24}$$

which follows by the complementarity condition of the LCP and the Cauchy–Schwartz inequality. Further, $\alpha > e^{T} w^{*} > \|w^{*}\|$ because of the non-negativity of the solution variables. Also $\varepsilon r < \frac{\varepsilon_5}{\alpha}$, so:

$$(w^{*})^{T} F(w^{*}) \leq \varepsilon_5. \tag{3.25}$$

To sum up the problem, (3.1)–(3.8) is solved by expanding the vectorial functions in a Taylor series around the iteration point and expressing the resulting linear complementarity problem approximation (3.18)–(3.20) of the given nonlinear complementarity problem within a suitable trust region.

Theorem 3.5. *The following are equivalent:*

1. *The nonlinear optimization problem defined by (3.1)–(3.8) has a solution,*
2. *The nonlinear complementarity problem defined by (3.9)–(3.11) has a solution.*
3. *The linear complementarity problem defined by (3.18)–(3.20) has a solution.*

Proof.

(1) → (2) : The nonlinear complementarity problem (3.9)–(3.11) is just a statement of Kuhn–Tucker necessary conditions for a solution of the nonlinear optimization (3.1)–(3.8),

(2) → (3) : Let the nonlinear complementarity problem (3.9)–(3.11) have a solution. This solution will satisfy the LCP (3.18)–(3.20),

(3) → (1) : Let the LCP (3.18)–(3.20) have a solution with the least number of barycentres, then it is a linearisation of the necessary Kuhn–Tucker conditions for a minimum solution to the nonlinear binary problem (3.1)–(3.8).

It has been shown that every linear complementarity problem can be solved by an appropriate parametric linear programming problem in a scalar variable [22]. The algorithm will find the solution of the linear complementarity problem, if such a solution exists, such that $\|w\| \leq \alpha$, for some constant $\alpha > 0$, or declare that no solution exists, so bounded. In this case the bound can be increased.

The termination of the classification algorithm may now be proved under a consistency condition.

Theorem 3.6. *Given a training set which does not contain two identical patterns assigned to different classes, then a correct classifier will be determined.*

Proof. If there is no juxtaposition of the patterns belonging to different classes, a feasible solution will always exist to the problem (3.1)–(3.8). Such a solution is to assign a unique barycentre to every pattern, with a resulting high value of the objective function.

Given that a feasible solution exists and that the objective function has a lower bound formed from the mean vectors to each class, an optimal solution to the problem (3.1)–(3.8) must exist.

From the results derived above by Theorem 3.5 the thesis follows.

3.4 Statistical Properties of the Classification Algorithm

Consider a training set defined over a suitable representation space, which is piecewise separable and coherent; therefore, the aim of this section is to determine the statistical properties that the set must satisfy so that it may be classified precisely by applying the algorithm **CASTOR** (**C**omplementarity **A**lgorithm **S**ystem for **TO**tal **R**ecognition [5]. A classification rule will apply to the dataset, and be just that partition which has been determined from the training set, so that to each entity in the dataset a class is assigned in line with the required properties. If the training set and the dataset which includes the training set forms a random sample, then this classification can be performed to any desired degree of accuracy by extending the size of the training sample. Sufficient conditions to ensure that these properties will hold if the dataset and the verification set are determined by non-repetitive random sampling. Consider therefore a dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where x_i is the feature vector of pattern i and its membership class is given by y_i.

Without loss of generality assume that classification problems of two classes only are considered, so that eventually a series of such problems must be solved for a polytomous classification problem. Assume, also, that the patterns are independently identically distributed with function $F(z)$, where $z_i = (x_i, y_i)$. Also let $f(x,\alpha) : \mathbb{R}^n \to \{0,1\}\ \alpha \in \Gamma$ be the classifier, where Γ is the set of parameters identifying the classification procedure from which the optimal parameters must be selected. The loss function of the classifier is given by:

$$L\left(y, f(x,\alpha)\right) = \left\{\begin{matrix} 0 & if & y = f(x,\alpha) \\ 1 & if & y \neq f(x,\alpha) \end{matrix}\right\} \tag{3.26}$$

The misclassification error over the population, in this case, is given by the risk functional:

$$R(\alpha) = \int L\left(y, f(x,\alpha)\right) \mathrm{d}F(x,y) \tag{3.27}$$

Thus, the value of $\alpha \in \Gamma$, say α^* must be chosen which renders the minimum expression (3.27). Hence, for any sample the misclassification error will be:

$$R^n_{\text{emp}}(\alpha^*) = \frac{1}{n}\sum_{i=1}^{n} L\left(y_i, f(x_i,\alpha^*)\right), \tag{3.28}$$

which will depend on the actual sample, its size n and the classifier used. To avoid introducing distributional properties on the dataset considered, the empirical risk minimization inductive principle may be applied, so that the risk functional $R(\alpha)$ given in (3.27) is replaced by the empirical risk functional $R^n_{\text{emp}}(\alpha)$ given by (3.28) constructed purely on the basis of the training set, and the function which minimizes risk is approximated by the function which minimizes empirical risk [31].

Definition 3.5. A dataset is stable, according to Definition 3.3, with respect to a partition and a population of entities if the relative frequency of misclassification is $R_{\text{emp}}\,(\alpha^*)\ \geq 0$ and

$$\lim_{n\to\infty} pr\{R_{\text{emp}}(\alpha^*) > \varepsilon\} = 0, \tag{3.29}$$

where α^* is the classification procedure applied, $\varepsilon > 0$ for given arbitrary small value and $pr\{.\}$ is the probability of the event included in the braces.

By considering smaller and smaller subsets of the attribute space X, if there exists a relationship between the attributes and the classes of the entities, the frequency of the entities of a given class for certain of these subsets will increase to the upper limit of one, while in other subsets it will decrease to a lower limit of zero. Thus for a very fine subdivision of the attribute space, each subset will tend to include entities only of a given class.

Definition 3.6. A proper subset S_k of the attribute space X of the dataset will give rise to a spurious classification if the conditional probability of a pattern that belongs to a given class c is equal to its unconditional probability over the attribute space. The dataset is spurious if this holds for all subsets of the attribute space X.

$$pr\{y_i = c \mid (y_i, x_i) \cap S_k\} = pr\{y_i = c \mid (y_i, x_i) \cap X\} \tag{3.30}$$

Theorem 3.7. *Consider a training set of n patterns randomly selected, assigned to two classes, where the unconditional probability of belonging to class one is p. Let a be a suitable large number and let $(n > a)$. Let the training set form b_n barycentres, then under* **CASTOR***, this training set will provide a spurious classification, if*

$$\frac{b_n}{n} \geq (1 - p) \qquad n > a \tag{3.31}$$

Proof. From the Definition 3.6, a classification is spurious if the class assigned to the entity is independent of the values of the set of attributes considered.

Any pattern will be assigned to the barycentre which is nearest to it, which without loss of generality may be considered a barycentre of class one, being composed of entities in class one. The probability that the pattern considered will result not of class one is $(1 - p)$, which is the probability that a new barycentre will be formed. As the number of patterns are n, the result follows.

Theorem 3.8. *Let the probability of a pattern to belong to class one be p, then the number of barycentres required to partition correctly a subset S, containing $n_s > a$ patterns, which is not spurious, formed from* **CASTOR** *algorithm is $b_s < n_s$, $\forall n_s > a$.*

Proof. If the classification is not spurious, by Definition 3.6, without loss of generality, the following relationship between the conditional and unconditional probabilities holds for one or more subsets $S_k, S_h \in X, S_h \cap S_k = \emptyset$:

$$pr\{y_i = 1 \mid (x_i, y_i) \cap S_k\} > pr\{y_i = 1 \mid (x_i, y_i) \cap X\} = p \tag{3.32}$$

$$pr\{y_i = 0 \mid (x_i, y_i) \cap S_h\} < pr\{y_i = 0 \mid (x_i, y_i) \cap X\} = (1 - p) \tag{3.33}$$

Thus on the basis of the algorithm, for the subsets $S_k \cap X$ the probability that a new barycentre of class one will be formed because one or more patterns result closer to a pattern of class zero is less than $(1 - p)$. In the set $S_h \cap X$, the probability that patterns of class one will appear is less than p, so that the probability that a pattern will be formed is less than p.

Therefore, if the number of patterns present in the subsets $S_k \cap X$ is n_k while the number of patterns present in the subsets $S_h \cap X$ is n_h, the total number of barycentres for the patterns of class one will be:

$$b_s < (1 - p)n_k + pn_h \tag{3.34}$$

As $n_s = n_k + n_h$, there results $b_s < n_s, \forall n_s > a$.

Corollary 3.2. *[31] The Vapnik–Cervonenkis dimension (VC dimension), $s(C, n)$ for the class of sets defined by the* **CASTOR** *algorithm restricted to the classification of a non-spurious dataset which is piecewise separable, with n_s elements, with two classes, is less than 2^{n_s}, if $n_s > a$.*

Proof. By Theorem 3.8, the number of different subsets formed is $b_s < n_s < 2^{n_s}$ whenever $n_s > a$ and the dataset is not spurious.

Theorem 3.9. [8] *Let C be a class of decision functions and ψ_n^* be a classifier restricted to the classification of a dataset which is not spurious and returns a value of the empirical error equal to zero based on the training sample $(z_1, z_2, \ldots, z_n)$. Thus, $Inf_{\psi \in C} L(\psi) = 0$, i.e. the Bayes decision is contained in C. Then,*

$$pr\,\{L(\psi_n^*) > \varepsilon\} \leq 2s(C, 2n)2^{\frac{-n\varepsilon}{2}} \tag{3.35}$$

By calculating bounds on the VC dimension, the universal consistency property can be established for this algorithm applied to the classification of a dataset which is not spurious.

Corollary 3.3. *[18] A non-spurious classification problem with a piecewise separable training set is strongly universally consistent.*

3.5 Experimental Results

In this implementation, the amino acid sequences of any protein is subdivided into segments of 13 amino acids. Each amino acid is coded as a five-bit string, and numbered from 1 to 20, so that each pattern vector is composed of 65 binary elements and the 66th element is assigned the class label of the amino acid corresponding to the class of the median one of the segment, element number 7. The training set used is indicated by the protein data bank (**PDB**) [3], which defines a pattern vector corresponding to that chosen segment of the amino acid sequence of the protein. No multiple alignment information is included. In the subsequent segment, 13 consecutive amino acids are considered, starting from the second one of the preceding segment and adding as a 13th segment the amino acid subsequent to the final one of the immediately previous segment defined. Two consecutive patters differ in the first and last element. Formally, a window of 13 amino acids is considered, and each pattern is formed by shifting the window of one position. Particular techniques are applied to initialise and terminate the patterns of a protein, and the class assigned to each pattern is always the folding class belonging to the seventh element in the pattern [5].

Consider all the sequences of the proteins which are included in a training set and compare them pairwise to determine the number of alignment amino acids common to the two proteins. An appropriate procedure is used to obtain the largest number of aligned amino acids by sliding the two sequences up and down and also inserting pieces of the string, according to strict rules [3]. For the similarity classification of the proteins in the training set, the largest alignment value is determined from the percentage of amino acids aligned between all proteins in the training set, and eight convenient classes of similitude are defined by setting suitable intervals of alignment percentage values. In Table 3.1, the similarity classes are shown together with the percentage interval of alignment scores or similitude which indicates interval of the largest percentage value of alignment of the protein in the training set.

For the purpose of this analysis, without loss of generality, proteins belonging to an isoform class are defined as proteins belonging to similarity class 7. This is taken as a necessary condition but is not a sufficient condition, since isoforms may have very different similarity, in which case the markers can be easily identified by traditional methods. Here, it is important to determine isoforms of proteins, which

Table 3.1 Similarity classes and percentage similarity among proteins

Similarity class	Similitude
0	<0.30
1	0.30–0.40
2	0.40–0.50
3	0.50–0.60
4	0.60–0.70
5	0.70–0.80
6	0.80–0.90
7	>0.90

have been identified as biomarkers, whose classification by traditional methods are insufficient to predict accurately the potential oncological malignities [15, 27, 34]. Thus, by identifying their secondary structure a segmentation of the set of proteins may be obtained to identify precise markers from the subsets of isoforms.

Training was performed for the dataset, considering a training set of over six million patterns and 2,500 proteins, which defines the verification set. Each amino acid in a segment was classified one by one to belong to a folding types α-helix, β-strand or coil. For every segment obtained from the dataset, the median amino acid was classified (or recognised in a verification set) so that as the median of each segment shifts by one residue, all the amino acids in the protein are iteratively recognised. The size of the segment could be modified to avoid structural diversity in identical subsequences, but in all the experiments carried out, the algorithm **CASTOR** converged in training to a completely correct assignment; from this information, it is believed that the behavior of the protein can be predicted more precisely and derive from the secondary structure the tertiary and quaternary structure, which is an ongoing research project. However, the prediction of the secondary structure can be used to characterise precisely the potential biomarkers to be considered.

To evaluate the algorithm proposed, a number of major procedures were used for comparison which are **PHD** [23]: **P**rofiled network from **He**i**d**elberg, **DSC** [16]: the **D**iscrimination of **S**econdary structure **C**LASS, **PRED** [11]: **PREDATOR**, **NNSSP** [36]: The **N**earest **N**eighbour **S**econdary **S**tructure **P**rediction, **MUL** (unpublished): The Mulpred algorithm **ZPRED** [38], and **CONS** [7]. All the classification algorithms used for comparison apply multiple alignments to the formation of the pattern vectors [23], since the inclusion of such information has long been recognized as a way to improve prediction accuracy. The increased precision obtained in most classification algorithms is 10–20% depending on the particular information used [14, 16]. However, the use of multiple alignments has been examined [5], and it is doubtful if this procedure is legitimate, or is rather biased, since the information on the protein to be classified may be contained in the alignment information and therefore the classification information is already known in part. The expected accuracy of the prediction procedures is higher than it should be, if this information were not considered. Per contra, the alignment information may help local classifications but if there are nonlinear interactions, the bias may give an incorrect result.

To test the result of the classification and comparison to other major classification algorithms, from the verification set of randomly selected patterns, those with similarity level of 7 present in all the indicated classification algorithm were selected as reported [7] and the results are given in Table 3.2. Moreover, in Table 3.2, for all the proteins in the verification set the standard classification precision measure was applied, indicated as Q_3 which can be calculated:

$$Q_3 = \frac{\sum_{i \in \{H,E,C\}} \text{number of residues predicted correctly}}{\sum_{i \in \{H,E,C\}} \text{number of residues in class } i} \tag{3.36}$$

where H stands for α-helix, E for β-strands, and C for coils. Each protein considered is defined by an alphanumeric label of four elements standard for all the

Table 3.2 Mean and Q_3 classification precision of the Cuff 513 verification set by similarity classes (56 proteins with **CASTOR** and traditional procedures selected)

Name	CASTOR	Sim.class	PHD	DSC	PRED	MUL	NNSSP	Zpred	CONS	Length
1eca	1.000	7	0.808	0.772	0.801	0.705	0.764	0.602	0.801	136
4rxn	1.000	7	0.648	0.611	0.611	0.611	0.648	0.574	0.666	54
1rbp	1.000	7	0.729	0.695	0.569	0.511	0.563	0.482	0.712	174
6tmn	1.000	7	0.585	0.651	0.639	0.579	0.607	0.585	0.620	316
4xia	0.988	7	0.778	0.743	0.732	0.722	0.773	0.692	0.778	393
1cel	0.983	7	0.651	0.623	0.637	0.584	0.637	0.579	0.658	433
1mns	0.982	7	0.714	0.636	0.723	0.614	0.675	0.561	0.714	228
1vnc	0.979	7	0.685	0.663	0.644	0.623	0.652	0.583	0.691	576
1cei	0.977	7	0.823	0.788	0.752	0.800	0.847	0.764	0.835	85
3chy	0.972	7	0.835	0.765	0.914	0.828	0.898	0.695	0.898	128
1com	0.971	7	0.798	0.689	0.647	0.588	0.638	0.579	0.773	119
1crn	0.956	7	0.413	0.587	0.456	0.413	0.456	0.391	0.456	46
1hxn	0.964	7	0.761	0.723	0.700	0.652	0.681	0.576	0.742	210
2cab	0.957	7	0.757	0.757	0.628	0.625	0.730	0.613	0.746	256
1cdl	0.909	7	0.750	0.950	0.450	0.300	0.800	0.250	0.850	20
1bam	0.896	7	0.600	0.605	0.585	0.490	0.715	0.510	0.675	200
1fc2	0.894	7	0.651	0.720	0.418	0.511	0.720	0.465	0.651	43
1cyx	0.865	7	0.765	0.721	0.727	0.607	0.645	0.645	0.765	158
2bop	0.852	7	0.600	0.564	0.670	0.529	0.423	0.552	0.635	85
1pyt	0.803	7	0.776	0.712	0.755	0.606	0.691	0.638	0.797	94
2asr	0.574	7	0.866	0.866	0.823	0.542	0.852	0.485	0.859	142
1pdo	0.500	7	0.860	0.790	0.790	0.736	0.821	0.643	0.852	129

major protein data bases and full details of everyone can be obtained from any of the databases. To analyze and determine precisely the secondary structure of the isoforms, an accurate classification algorithm **CASTOR** was applied.

The predicted secondary structure is accurate and higher than the one obtained by other procedures. The only exceptions are for 2 proteins 2asr (chemotaxis: the three-dimensional structure of the aspartate receptor from *Escherichia coli*) and 1pdo (phosphotransferase: phosphoenolpyruvate-dependent phosphotransferase system). Considering that there are 2,506 protein patterns in the verification set, and 1,556 result in the similarity class 7, the mean prediction accuracy is 87%, while the $\mathbf{Q}_3$ mean measure resulted 88% [5].

From Table 3.2, the results indicate that the secondary structure can be determined from the sequences irrespective of the species and the type of cell considered, since fundamental proteins such as histones or ribosomes and other types were all correctly classified. The innovatory aspect of this research is that the proteins of class 7 include far more different protein families than just structural and enzymatic proteins. Identification of the secondary structure might therefore lead to a novel method to study biological functions in addition of a method to predict precisely oncological malignancies, by defining appropriate markers. From the theory developed in the previous sections and the experimental results given elsewhere [5], it should be evident that proteins that arise infrequently are recognizable precisely, and their analysis should allow to distinguish the most appropriate typology of similar proteins.

Table 3.3 Mean and Q_3 classification precision of isoforms from the Cuff 513 verification set by similarity classes (56 proteins with **CASTOR** and traditional procedures selected)

Name	CASTOR	Sim.class	PHD	DSC	PRED	MUL	NNSSP	Zpred	CONS	Length
1gal$_2$	0.907	7	0.591	0.569	0.489	0.462	0.639	0.500	0.612	186
1gal$_3$	0.981	7	0.698	0.698	0.689	0.689	0.741	0.637	0.732	116
1scu_1	0.917	7	0.768	0.743	0.727	0.677	0.719	0.661	0.752	121
1scu_2	0.938	7	0.802	0.777	0.777	0.691	0.827	0.654	0.814	81
1scu_3	0.967	7	0.845	0.765	0.812	0.798	0.879	0.758	0.879	149
2dln$_1$	0.918	7	0.616	0.547	0.726	0.698	0.575	0.561	0.643	73
2dln$_3$	0.962	7	0.678	0.714	0.678	0.666	0.726	0.547	0.714	84
2adm	0.962	7	0.538	0.615	0.526	0.485	0.603	0.479	0.597	169
2adm	0.962	7	0.652	0.643	0.754	0.537	0.680	0.574	0.685	216
1dpg$_1$	0.961	0	0.875	0.711	0.779	0.717	0.830	0.694	0.875	177
1dpg$_2$	0.902	0	0.730	0.633	0.642	0.659	0.665	0.594	0.698	308
1rec$_1$	0.812	3	0.686	0.735	0.696	0.705	0.754	0.686	0.715	102
1rec$_2$	0.907	3	0.783	0.771	0.819	0.759	0.867	0.650	0.843	83

Table 3.3 shows the classification precision of **CASTOR** for six different families of proteins and, again, the higher accuracy of this classification algorithm compared with other procedures is noted. 1Gal is a flavoprotein oxydoreductase of the glucose constituted by 581 amino acids involved in the respiratory chain of the energetic pathways of the cells evidenced in the **PDB** database in two slightly different forms: 1Scu is an ATP-binding protein tetramer, a ligase, also known as succinyl-CoA synthetase. Its catalytic activity is involved in metabolic processes. 2Dln is a ligase with a protein chain of 306 residues involved in the biosynthesis of peptidoglycans of the cell wall in *Escherichia coli*. Further, 2Adm is a methyltransferase that counts 385 amino acids that catalyzes methylations involved in nucleic acid binding. 1Dpg is an oxydoreductase constituted by a dimer (485 amino acids) that enters in the early stages of the pentose phosphate pathway. Finally, 1rec (or recoverin) is a calcium-binding protein with 185 amino acids that belongs to the EF hand superfamily; it serves as a calcium sensor in vision.

In Table 3.4, the small differences in the primary structure which give rise to alternative sequences in the database can be considered as isoforms of the basic protein, although their length may be quite different.

For this algorithm, the variation in the precision of the classification among given classes of isoforms is very limited and the overall value is high, within experimental variation. The secondary structure may differ among the isoforms of a protein and precise structures are determined.

The folding characteristics of the protein and its isoforms are reported. The database indications of the secondary structure were obtained from the database specified. In many cases, the protein considered in the database is different from the isoforms reported, obtained from the **PDB** database, hence all the estimated folding characteristics obtained with **CASTOR** algorithm should be compared with the results indicated in the databases.

Table 3.4 Folding characteristics of the protein given in EMBL-EBI and isoforms entries in the **PDB** databases according to the **CASTOR** algorithm selected)

Protein/isoform	Length	Helix	Strand	Other (coil)
1gal	581	54	26	66
1gal$_2$	186	19	7	23
1gal$_3$	116	10	7	18
1scu[a]	676	61	39	68
1scu$_1$	121	9	7	11
1scu$_1$	81	9	6	12
1scu$_1$	149	9	9	15
2dlin$_1$	306	23	16	32
2dlin$_1$	73	7	6	8
2dlin$_3$	84	7	9	9
2adm	385	29	27	47
2adm	169	15	13	31
2adm	216	14	16	21
1dpg$_1$	485	50	17	19
1dpg$_1$	177	16	9	7
1dpg$_2$	308	32	12	11
1rec$_1$	185	35	6	7
1rec$_1$	102	29	4	6
1rec$_1$	83	7	5	5

[a] 2 chains of proteins included

The isoform folding characteristics are for the major part not given in the databases; therefore, to provide a partial comparison, the only given folding values for the main protein is compared to the results determined with the **CASTOR** algorithm.

Further analysis should be made to determine exactly the differences that occur, and this research will be available in the near future. Nevertheless, the results here indicated are deemed important as it is evident, even with these partial results, that the secondary structure of the isoforms differ and their functional characteristics will differ.

Thus, it is useful to determine the secondary structure of proteins and isoforms so that the markers identified have unique characteristics and can be used as a standard for diagnosis.

Further the prediction of secondary structure among isoforms may lead to a better comprehension of the biological effect of mutations among the isoforms of a protein. This might permit to determine more accurately the relationship of the biological functions of isoforms and the protein of the class.

The analysis of secondary structures might lead to the identification of subtypes among oncologic markers, defining also their biological implication. It is known that subtypes of markers already identified are related to more precise prognosis and to the eventual clinical evolution of the malignancy. The differences between two isoforms may be as small as just one amino acid with, at the moment, totally unpredictable effects on life, unless all the functional dependencies are fully determined.

This approach permits the effects to be determined more precisely, even if there are only small modifications in the primary structure.

There are many individual risk factors that have been identified for breast cancer. Mutations within BRCA1 or BRCA2 have been considered, and the majority of inherited breast cancer are attributed to these factors [10].

The study of hypoxic factors in BRCA1, BRCA2, and BRCAX breast cancers has been carried out to study clinicopathological parameters and the intrinsic breast cancer phenotypes. BRCA1 tumors correlated with basal phenotypes, various receptors, and absence of lymph node metastasis. The aggressive nature of BRCA1 and basal-type tumours may be partly explained by an enhanced hypoxic drive and hypoxia-inducible factors (HIF) driven degradations because of suppressed and aberrantly located FIH (factor inhibiting HIF) expressions. This may have important implications, as these tumours may respond to compounds directed against certain factors [35].

Since the gene encoding BRCA1 was first cloned in 1994, researchers have sought to establish the molecular basis for its linkage to breast and ovarian cancer. As universal functions for this protein have emerged, questions persist concerning how its disruption can elicit cancer in a tissue- and gender-specific manner. A functional interrelationship between BRCA1 and estrogen signaling may be involved in breast tumorigenesis [30].

There are at least five isoforms of the protein BRCA which play an important role in breast cancer pathologies, specified as BRCA1, BRCA2, BRCA3, BRCA4, and BRCA5. However, the function of each isoform is not clear as indicated above and so the identification of each isoform, which is determined by electrophoresis or other probes or protein microarrays, should be integrated by the secondary structure to make the isoform unique with respect to the functional characteristics.

In Table 3.5, the dimensions and the properties of the sequence alignment are given. Comparison of the primary sequence of the isoforms by pair indicates extensive variations concerning the properties of the residues regarding the identity of residues, their similarity, and the recourse to gaps to obtain the highest value of similarity.

The comparative similarity of the folding characteristics for any pair of isoforms can differ extensively even for similar isoforms since the type, gaps to obtain the maximum similarity, and number of different residues must be considered.

The application of the **CASTOR** algorithm on the primary structure of each isoform yields the folding structure indicated. In particular, it is important to note that although BRCA1 and BRCA2 differ in one residue, the effect of this modification yields a markedly different folding structure, which may provide part of the justification for the different effects on the malignancies of these isoforms.

The biological function of the BRCA gene products indicated have been studied for over a decade, and the predisposition of their role in cancer should be studied [32]. On the other hand, it is reaffirmed that the significance prevalence of hereditary breast cancer in women (both BRCA1- and BRCA2- associated disease) plays an important role. Moreover, the triple-negative immunophenotype is an imperfect surrogate measure of germlike BRCA status [2].

In ulterior important research endeavours, the study of the pathways of the malignancies and the management of the pathologies associated [12, 17] are pursued with respect to many different factors that may influence the pathologies. However

Table 3.5 Folding characteristics of the protein given in EMBL-EBI and isoforms entries in the PDB databases according to the **CASTOR** algorithm

Isoform	BRCA1	BRCA2	BRCA3	BRCA4	BRCA5
Length	1,860	1,860	760	1,846	720
BRCA1 – Identity %	–	99.9	40.6	98.9	38.6
– Similarity %		99.9	40.6	98.9	38.6
– Gaps %		0.0	59.4	1.1	61.4
– Differences residues N		1	–	20	–
BRCA2 – Identity %	–	–	40.6	98.9	37.7
– Similarity %		–	40.6	98.9	37.7
– Gaps %		–	59.4	1.1	62.3
– Differences residues N		–	–	20	–
BRCA3 – Identity %		–	–	39.9	94.7
– Similarity %		–	–	39.9	94.7
– Gaps %		–	–	60.1	5.3
– Differences residues N		–	–	–	40
BRCA4 – Identity %		–	–	–	37.7
– Similarity %		–	–	–	37.7
– Gaps %		–	–	–	62.3
– Differences residues N		–	–	–	–
Folding segments					
Helix folds	167	164	71	147	78
Strands folds	82	83	38	72	41
Other folds	199	198	69	177	73

in many cases these factors may be multi-determined, and the complete analysis of these aspects should be studied. There may be some important dependencies between different factors, so the effect of a modification may be undetermined, and this will give rise to random-like effects, as reported in many studies.

A more complete analysis and a precise formalization of the various factors that may affect malignancies should be pursued. This will require a detailed study of the phenomenon so that the predictions of the outcome can be determined with a given precision, and therefore, there will arise a certain confidence in the role of each aspect.

To carry out these types of studies, the utilization of the prediction of the secondary structure of the proteins should be considered as a useful stage toward the recourse to tertiary and quaternary structures and a more precise formulation of the factors to be examined.

3.6 Conclusions

A classification process called CASTOR is described. It is based on a nonlinear binary optimization algorithm as an implementation of a model which is derived on a strict statistical methodology avoiding extraneous assumptions, which could limit the precision. Thus, through this implementation the secondary structure of some isoform proteins can be identified and recognised as biomarkers to oncological pathologies. This study has examined the potential benefits of secondary structure predictions of proteins to permit a more precise formulation of isoforms and consider the differential effects of each alternative isoform on the outcome.

To determine protein markers in oncology, the isoforms of possible proteins must be distinguished and then their incidence on the pathology must be ascertained. For instance, various BRCA isoforms may be determined correctly, but the selection process could hide variants of some of these isoforms, as is evident in the results presented, which may have differential effects on breast cancer.

References

1. P. Alvarez-Chaver, A. M. Rodriguez-Pineiro, F. J. Rodririguez-Berrocal, V. S. Martinez-Zorzano, and M. Paez de la Cadena. Identification of hydorphobic proteins as biomarker candidates for colorectal cancer. *International Journal of Biochemistry and Cell Biology*, 39:529–540, 2007
2. D. P. Atchley, C. T. Albarracin, and A. Lopez et al. Clinical and pathologic characteristics of patients with brca-positive and brca-negative breast cancer. *Breast Diseases: A Year ool Quaterly*, 20:145–146, 2009
3. H. M. Berman, J. Westbrook, Z. Feng, G. Gilarid, T. N. Bhat, H. Wessig, I. N. Shindyakov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000
4. S. Buhlmann and B. M. Pützer. Dnp73 a matter of cancer: Mechanisms and clinical implications. *Biochimica et Biophysica Acta*, 178:207–216, 2008
5. C. Cifarelli and G. Patrizi. Solving large protein secondary structure classification problems by a nonlinear complementarity algorithm with 0,1 variables. *Optimization Methods and Software*, 22:25–49, 2007
6. B. I. Cohen, S. R. Presnell, and F. E. Cohen. Origins of structural diversity within sequentially identical hexapeptides. *Protein Science*, 2:2134–2145, 1993
7. J. A. Cuff and G. J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure Functions and Genetics*, 34:508–519, 1999
8. L. Devroye, L. Gyorfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, Berlin, 1996
9. L. Di Giacomo, E. Argento, and G. Patrizi. Linear complementarity methods for the solution of combinatorial problems. *Informs Journal of Computing*, 19:73–79, 2007
10. J. P. Evans, C. Skrzynia, L. Susswein, and M. Harlan. Genetics and the young woman with breast cancer. *Breast Disease*, 23:17–29, 2009
11. D. Frishman and P. Argos. Knowledge-based protein secondary structure assignment. *Proteins: Structure Functions and Genetics*, 23:566–579, 1995
12. M. J. Garcia and J. Benitez. The fanconi anaemia/brca pathway and cancer sussceptibility, searching for new therapeutic targets. *Clinical Transactional Oncology*, 10:78–84, 2008
13. N. Jardine and R. Sibson. *Mathematical Taxonomy*. Wiley, New York, 1971
14. D. T. Jones. Protein structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292(2):195–202, 1999
15. A. Kawai, T. Kondo, Y. Suchara, K. Kikuta, and S. Hirobashi. Global proptein-expression analysis of bone and soft tissue sarcomas. *Clinical Orthopaedics and Related Research*, 466:2099–2106, 2008

16. R. D. King and M. J. E. Stenberg. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Protein Science*, 5(11): 2298–2310, 1996
17. F. P. Liebens, B. Carly, A. Pastijn, and S. Rozenberg. Management of brca1/2 associated breast canncer: a systematic qualitative review of the state of knowledge in 2006. *European Journal of Cancer*, 43:138–157, 2007
18. L. Nieddu and G. Patrizi. Formal properties of pattern recognition algorithms: A review. *European Journal of Operational Research*, 120:459–495, 2000
19. J.-S. Pang F. Facchinei. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer, Berlin, 2003
20. E. V. Pankratova. Alternative promoters in expression of genetic informatio. *Molecular Biology*, 42:422–433, 2008
21. G. Patrizi. Optimal clustering properties. *Ricerca Operativa*, 10:41–64, 1979
22. G. Patrizi. The equivalence of an lcp to a parametric linear program with a scalar paramter. *European Journal of Operational Research*, 51:367–386, 1991
23. B. Rost and C. Sander. Prediction of secondary structure at better than 70% accuracy. *Journal of Molcular Biology*, 232:584–599, 1993
24. P. Rubiczey, A. Tordai, H. Andrikovics, A. G. Filoteo, J. T. Penniston, J. Enouf, A. Enyedi, B. Papp, and T. Kovacs. Isoform-specific up-regulation of plasma membrane Ca^{2+} ATPase expression during colon and gastric cancer cell differentiation. *Cell Calcium*, 42:590–605, 2007
25. B. W. Schafer and C. W. Heizman. The s100 family of ef-hand calcium-binding proteins: Functions and pathology. *Trends in Biochemical Sciences*, 21:134–140, 1996
26. L. Silvestri and J. R. Hill. Some problems on the taxonomic approach. In V. H. Heywood and J. McNeill, editors, *Phonetic and Philogenic Classification*, pages 87–104. Systematics Association, London, 1964
27. A. Stangelberger, M. Margreiter, C. Seitz, and B. Djavan. Prostate cancer screening makers. *The Journal of Men's Health & Gender*, 4:233–244, 2007
28. S. Sudarsanam. Structural diversity of sequentially identical subsequences of proteins: Identical octapeptides can have different conformations. *Proteins: Structure, Function and Genetics*, 30:228–231, 1998
29. I. E. Tothill. Biosensors for cancer markers diagnosis. *Seminars in Cell and Development Biology*, 20:55–62, 2009
30. A. M. Trauernicht and T. G. Boyer. Brca1 and estrogen signaling in breast cancer. *Breast Disease*, 18:11–20, 2009
31. V. N. Vapnik. *Learning Theory*. Wiley, New York, 1998
32. A. R. Venkitaraman. Targeting the molecular defects in brca-deficient tumors for cancer therapy. *Cancer Cell*, 16:89–90, 2009
33. S. Watanabe. *Pattern Recognition: Human and Mechanical*. Wiley, New York, 1985
34. R. Wolf, C. Vascopoulos, J. Winston, A. Dharamsi, P. Goldsmith, M. Gunsior, B. K. Vonderhaar, M. Olson, and P. H. Watson. Highly homologous hs100a15 and hs100a7 proteins are distinctly expressed in normal breast tissue and breast cancer. *Cancer Letters*, 277:101–107, 2009
35. M. Yan, M. Rayoo, E. A. Takano, Investigations KConFab, and S. B. Fox. Brca1 tumors correlate with a HIF-1alpha phenotype and have a poor prognosis through modulation of hydroxylase enzyme profile expression. *British Journal of Cancer*, 101:1168–1174, 2009
36. T. M. Yi and E. S. Lander. Protein secondary structure prediction using nearest-neighbour methods. *Journal of Molecular Biology*, 225:1049–1063, 1993
37. A. Zajac, D. Song, W. Qian, and T. Zhakov. Protein microarrays and quantum dot probes for early cancer detection. *Colloids and Surfaces B*, 58:309–314, 2007
38. M. J. J. M. Zvelebil, G. J. Barton, W. R. Taylor, and M. J. E. Sternberg. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *Journal of Molecular Biology*, 195:957–961, 1987

Chapter 4
Protein Fold Recognition Using Markov Logic Networks

Marenglen Biba, Stefano Ferilli, and Floriana Esposito

Abstract Protein fold recognition is the problem of determining whether a given protein sequence folds into a previously observed structure. An uncertainty complication is that it is not always true that the structure has been previously observed. Markov logic networks (MLNs) are a powerful representation that combines first-order logic and probability by attaching weights to first-order formulas and using these as templates for features of Markov networks. In this chapter, we describe a simple temporal extension of MLNs that is able to deal with sequences of logical atoms. We also propose iterated robust tabu search (IRoTS) for maximum a posteriori (MAP) inference and Markov Chain-IRoTS (MC-IRoTS) for conditional inference in the new framework. We show how MC-IRoTS can also be used for discriminative weight learning. We describe how sequences of protein secondary structure can be modeled through the proposed language and show through some preliminary experiments the promise of our approach for the problem of protein fold recognition from these sequences.

4.1 Introduction

Protein fold recognition is the problem of determining whether a given protein sequence folds into a previously observed structure. An uncertainty complication is that it is not always true that the structure has been previously observed. Therefore, there is strong motivation for developing machine learning methods that can automatically infer models from already observed sequences in order to classify new instances.

Dealing with sequential data has become an important application area of machine learning. Such data are frequently found in computational biology, speech recognition, activity recognition, information extraction, etc. One of the main

M. Biba (✉)
Department of Computer Science, University of New York, Tirana, Albania
e-mail: marenglenbiba@unyt.edu.al

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_4,

problems in this area of machine learning is assigning labels to sequences of objects. This class of problems has been called *sequential supervised learning* [7]. Probabilistic graphical models and in particular hidden Markov models (HMM) have been quite successful in modeling sequential phenomena. However, the main weaknesses for this model are: (1) It handles sequences of flat alphabets only (i.e., sequential objects have no structure) and (2) It is hard to express dependencies in the input data. Recently, to overcome the first problem, the work in [16] introduced logical hidden Markov models (LoHMM), an extension of HMM to handle sequences of logical atoms. However, the second problem still remains for LoHMM. For this reason, conditional random fields (CRFs) [12] have been proposed. CRFs are discriminatively trained graphical models instead of generatively trained such as HMMs. CRFs can easily handle non-independent input features and represent the conditional probability distribution $P(Y|X)$, where X represents elements of the input space and Y of the output space. For many tasks in computational biology, information extraction or user modeling CRF have outperformed HMMs.

One of the problems where sequences exhibit internal structure is modeling sequences of protein secondary structure. These sequences can be seen as sequences of logical atoms (details about logic can be found in [8]). For example, the following sequence of the TIM beta/alpha-barrel protein represents a sequence of logical atoms:

$$st('SB', null, medium), st('SB', plus, medium), he(h(right, alpha), long),$$

$$st('SB', plus, medium), he(h(right, alpha), medium), ...$$

Helices and strands are represented, respectively, by *he(type,length)* and *st(orientation, length)*. Traditional HMMs or CRFs would ignore the structure of the symbols in the sequence loosing therefore the structure that each symbol implies or would take into account all the possible combinations (of orientation and length) into account that could lead to a combinatorial explosion of the number of parameters.

The first approach to dealing with sequences of logical atoms by extending CRFs is that of [10] where the authors propose TildeCRF that uses relational regression trees in the gradient tree boosting approach [7] to make relational abstraction through logical variables and unification. The authors showed that TildeCRF outperformed previous approaches based on LoHMMs such as [6, 10].

Many real-world application domains are characterized by both uncertainty and complex relational structure. Statistical learning focuses on the former, and relational learning on the latter. Statistical relational learning [9] aims at combining the power of both. One of the representation formalisms in this area is Markov Logic which subsumes both finite first-order logic and probabilistic graphical models as special cases [23]. Upon this formalism, Markov logic networks (MLNs) can be built serving as templates for constructing Markov networks (MNs). In Markov Logic a weight is attached to each clause and learning an MLN consists of structure learning (learning the clauses) and weight learning (setting the weight of each clause).

In this chapter, we describe Stochastic MLNs, a simple model based on MLNs that is able to deal with sequences of logical atoms. We also propose two algorithms for inference and learning in SMLNs based on the iterated robust tabu search metaheuristic. Finally, we model in SMLNs the problem of protein fold recognition from sequences of protein secondary structure and show through some preliminary experiments the promise of our approach.

The chapter is organized as follows: in Sect. 4.2 we introduce MNs and MLNs, in Sect. 4.3 we describe existing learning approaches for MLNs, Sect. 4.4 introduces Stochastic MLNs, Sect. 4.5 introduces the iterated local search (ILS) and robust tabu search (RoTS) metaheuristic and the satisfiability solver iterated robust tabu search (IRoTS) that combines both and describes how IRoTS can be used for Maximum a posteriori (MAP) inference in MLNs instead of the Viterbi algorithm, Sect. 4.6 introduces Markov chain IRoTS (MC-IRoTS) for performing inference in MLNs, Sect. 4.7 describes how protein sequences can be modeled in SMLNs. Section 4.8 presents some preliminary experiments, and Sect. 4.9 concludes with future work.

4.2 Markov Networks and Markov Logic Networks

A Markov Network (or Markov random field) represents a model for the joint distribution of a set of variables X = (X_1,X_2,...,X_n) $\in \chi$ [5] (in this chapter, we deal only with discrete features and variables). The model is composed of an undirected graph G and a set of potential functions. There is a node for each variable and a potential function ϕ_k for each clique in the graph (a clique in an undirected graph G is a set of vertices V, such that for every two vertices in V, there exists an edge connecting the two). A potential function is a non-negative real-valued function of the state of the corresponding clique. The joint distribution defined by an MN is given by the following:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}), \tag{4.1}$$

where $x_{\{k\}}$ is the state of the kth clique (i.e., the state of the variables that appear in that clique). The partition function, denoted by Z, is:

$$Z = \sum_{x \in \chi} \prod_k \phi_k(x_{\{k\}}) \tag{4.2}$$

MNs are often represented as log-linear models, by replacing each clique potential with an exponentiated weighted sum of features of the clique state. This replacement gives the following:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_j w_j f_j(x)\right) \tag{4.3}$$

A feature f may be any real-valued function of the state. The focus of this chapter is on binary features, $f_j \in \{0, 1\}$. Thus, if we translate from the potential-function form, the model will have one feature for each possible state x_k of each clique and its weight will be $\log(\phi(x_{\{k\}})$. This representation is exponential in the size of the cliques, but, however, we can specify a much smaller number of features in a more compact representation than the potential-function form. This is the case when large cliques are present and MLNs try to take advantage of this.

A first-order knowledge base (KB) can be considered as a set of hard constraints on a set of possible worlds: if a world violates a single formula, it will have zero probability. The idea in Markov logic is to soften these constraints: when a world violates a formula in the KB it will be less probable, but not impossible. The fewer formulas a world will violate, the more probable it will be. Each formula has an attached weight that represents how hard a constraint it is. A higher weight of a formula means there is a greater difference in log probability between a world that satisfies that formula and one that does not, all other things being equal.

An MLN [23] T is a set of pairs $(F_i; w_i)$, where F_i is a formula in first-order logic (FOL) and w_i is a real number. Together with a finite set of constants $\mathrm{C} = \{c_1, c_2, \ldots, c_p\}$ it defines a MN $M_{\mathrm{T;C}}$ as follows:

1. There is a binary node in $M_{\mathrm{T;C}}$ for each possible grounding of each predicate appearing in T and the value of the node is 1 if the ground predicate is true, and 0 otherwise.
2. There is one feature in $M_{\mathrm{T;C}}$ for each possible grounding of each formula F_i in T and the value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight w_i of the formula F_i in T becomes the weight of this feature. There is an edge between two nodes of $M_{\mathrm{T;C}}$ if and only if the corresponding ground predicates appear together in at least one grounding of a formula in T.

An MLN can be viewed as a template for constructing MNs. The probability distribution over possible worlds x defined by the ground MN $M_{\mathrm{T;C}}$ is given by:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^{F} w_i n_i(x)\right), \tag{4.4}$$

where F is the number of formulas in T and $n_i(x)$ is the number of true groundings of F_i in x. When formula weights increase, an MLN will resemble a purely logical KB, and in the limit of all infinite weights it becomes equivalent to it.

The focus of this chapter is on MLNs with function-free clauses assuming domain closure in order to ensure that the MNs generated will be finite. In this case, the groundings of a formula are formed by replacing the variables with constants in all possible ways.

A simple example of a first-order KB is given in Fig. 4.1. Statements in FOL are always true. The following FOL formulas state that if someone drinks heavily, he will have an accident, and that if two people are friends, they either both drink or both don't drink.

$$\forall x \quad HeavilyDrinks(x) \Rightarrow CarAccidents(x)$$

$$\forall x, y \quad Friends(x,y) \Rightarrow (HeavilyDrinks(x) \Leftrightarrow HeavilyDrinks(x))$$

Fig. 4.1 Example of a knowledge base in first-order logic

$$1.8 \quad \forall x \; HeavilyDrinks(x) \Rightarrow CarAccidents(x)$$

$$0.7 \quad \forall x, y \; Friends(x,y) \Rightarrow (HeavilyDrinks(x) \Leftrightarrow HeavilyDrinks(x))$$

Fig. 4.2 Example of a knowledge base in Markov logic

$$1.8 \quad \forall x \; HeavilyDrinks(x) \Rightarrow CarAccidents(x)$$

$$0.7 \; \forall x, y \; Friends(x,y) \Rightarrow (HeavilyDrinks(x) \Leftrightarrow HeavilyDrinks(x))$$

Constants: Paolo (P) and Cesare (C)

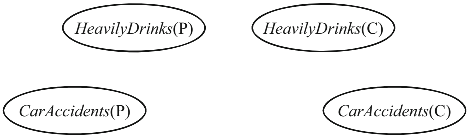


Fig. 4.3 Partial construction of the nodes of the ground Markov network

Since FOL statements, in practice are not always true, it is necessary to soften these hard constraints. For example, in practice it is not always true that if someone drinks heavily, he will have a car accident. In Fig. 4.2, it is presented a KB in Markov Logic. As it can be seen, formulas have weights attached and statements are not always true any more. Their degree of truth depends on the weight attached. For instance, the first formula expresses a stronger constraint than the second.

The simple KB in Fig. 4.2 together with a set of constants defines an MN. For example, suppose we have two constants in the domain that represent two persons, Paolo and Cesare. Then, the first step in the construction on the MN is given by the grounding of each predicate in the domain according to the constants of the domain. Partial grounding is shown in Fig. 4.3 where only groundings of *HDrinks* and *CarAcc* are considered. The complete nodes are shown in Fig. 4.4 where all the groundings of the predicates represent nodes in the graph.

In the next step, any two nodes whose corresponding predicates appear together is some ground formula are connected. For example, in Fig. 4.5, the nodes *HDrinks(P)* and *CarAcc(P)* are connected through an arc, because the two predicates appear together in the grounding of the second formula. The complete graph is presented in Fig. 4.6.

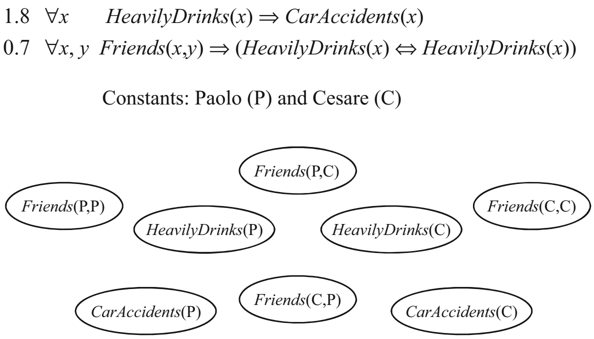

1.8 ∀x HeavilyDrinks(x) ⇒ CarAccidents(x)
0.7 ∀x, y Friends(x,y) ⇒ (HeavilyDrinks(x) ⇔ HeavilyDrinks(x))
Constants: Paolo (P) and Cesare (C)
Friends(P,C)
Friends(P,P)
HeavilyDrinks(P)
HeavilyDrinks(C)
Friends(C,C)
CarAccidents(P)
Friends(C,P)
CarAccidents(C)

Fig. 4.4 Complete construction of the nodes of the ground Markov network

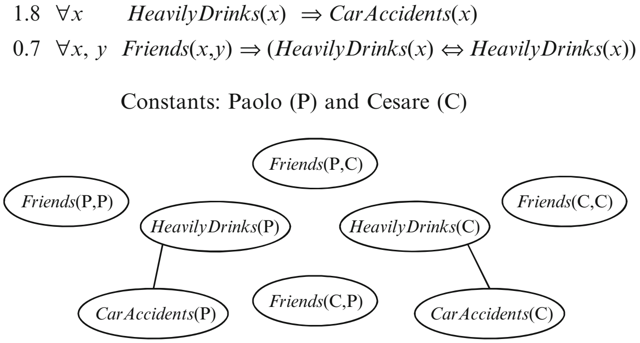

1.8 ∀x HeavilyDrinks(x) ⇒ CarAccidents(x)
0.7 ∀x, y Friends(x,y) ⇒ (HeavilyDrinks(x) ⇔ HeavilyDrinks(x))
Constants: Paolo (P) and Cesare (C)
Friends(P,C)
Friends(P,P)
HeavilyDrinks(P)
HeavilyDrinks(C)
Friends(C,C)
CarAccidents(P)
Friends(C,P)
CarAccidents(C)

Fig. 4.5 Connecting nodes whose predicates appear in some ground formula

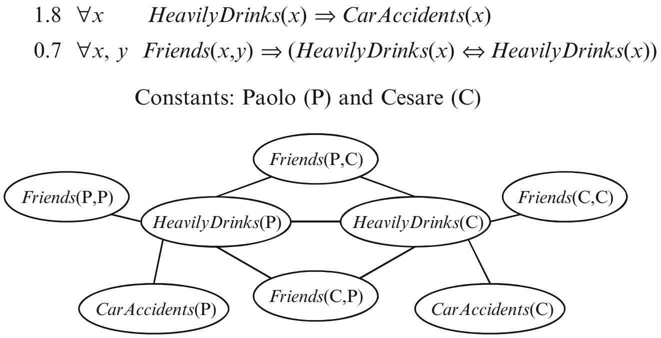

1.8 ∀x HeavilyDrinks(x) ⇒ CarAccidents(x)
0.7 ∀x, y Friends(x,y) ⇒ (HeavilyDrinks(x) ⇔ HeavilyDrinks(x))
Constants: Paolo (P) and Cesare (C)
Friends(P,C)
Friends(P,P)
HeavilyDrinks(P)
HeavilyDrinks(C)
Friends(C,C)
CarAccidents(P)
Friends(C,P)
CarAccidents(C)

Fig. 4.6 Connecting nodes whose predicates appear in some ground formula

4.3 Learning Approaches for MLNs

The first attempt to learn MLNs structure was that in [23], where the authors used an inductive logic programming (ILP) system to learn the clauses and then learned the weights by maximizing pseudo-likelihood [1]. In [17] another method was proposed that combines ideas from ILP and feature induction of Markov networks. This algorithm, which performs a beam or shortest first search in the space of clauses guided by a weighted pseudo-log-likelihood (WPLL) measure, outperformed that of [23]. Recently, in [14] a bottom-up approach was proposed in order to reduce the search space. This algorithm uses a propositional Markov network learning method to construct template networks that guide the construction of candidate clauses. In this way, it generates fewer candidates for evaluation. Recently, a structure learning algorithm based on ILS was proposed in [2] and it was shown to improve over those in [14, 17]. For every candidate structure, in all these algorithms the parameters that optimize the WPLL are set through limited-memory BFGS algorithm [30]. L-BFGS approximates the second derivative of the WPLL by keeping a running finite-sized window of previous first derivatives. Another algorithm that works in a discriminative fashion was proposed in [3] which scores structures by conditional likelihood and learns parameters by maximum likelihood.

Learning MLNs in a discriminative fashion has produced much better results for predictive tasks than generative approaches as the results in [25] show. In this work, the voted-perceptron algorithm was generalized to arbitrary MLNs by replacing the Viterbi algorithm with a weighted satisfiability solver. The voted perceptron is a special case in which tractable inference is possible using the Viterbi algorithm [4]. The new algorithm is gradient descent with a most probable explanation (MPE) approximation to the expected sufficient statistics (true clause counts). These could vary widely between clauses, causing the learning problem to be highly ill-conditioned and making gradient descent very slow. In [15] a preconditioned scaled conjugate gradient approach is shown to outperform the algorithm in [25] in terms of learning time and prediction accuracy. This algorithm is based on the scaled conjugate gradient method and very good results are obtained with a simple approach: per-weight learning weights, with the weight's learning rate being the global one divided by the corresponding clause's empirical number of true groundings. The scaled conjugate gradient approach was originally proposed in [31] for training neural networks.

4.4 Temporal Extension of Markov Logic

In stochastic processes, the world evolves and a ground predicate's truth value depends on the time step t. In order to model in MLNs the evolution of objects and relations, we need to represent time. To achieve this, we introduce the concept of *temporal predicates* which have an additional time argument. Time is represented as a non-negative integer variable, and all predicates are of the form $P(x_1,\ldots,x_n,t)$ where t denotes time.

We define stochastic Markov logic networks (SMLNs) as a set of MLN formulas defined on the *temporal predicates*. In addition, we borrow from linear temporal logic (LTL) [19] the concept of *temporal operator* and introduce *succ*, a *time predicate* (similar to the operator *next* in LTL) that represents the successor of a time step t, i.e., succ(1,2), succ(2,3), and so on.

In principle, the structure of SMLNs could be learned in a similar fashion as for MLNs. This task could be made easier by imposing further restrictions such as the Markovian assumption, or through the use of *declarative bias*. This can be performed using the structure learning algorithms proposed in [2,3]. In this chapter, we assume the structure of the model to be fixed and try to learn optimal parameters for discriminative training of SMLNs for the problem of protein fold recognition from sequences of secondary structure.

4.5 MAP Inference Using Iterated Robust Tabu Search

In logic, one of the main problems is determining whether a KB is satisfiable (SAT problem), i.e., if there is a truth assignment to ground atoms that make the KB true. This problem is NP-complete. However, stochastic local search (SLS) methods have made great progress toward efficiently solving SAT problems with hundreds of thousands of variables in a few minutes. The optimization variant of SAT is MAX-SAT where the problem is to find a truth assignment that maximizes the number of satisfied clauses (unweighted MAX-SAT) or if clauses have an associated weight, maximize the total weight of the satisfied clauses (weighted MAX-SAT). Both, unweighted and weighted MAX-SAT are NP-complete problems. First-order problems can also be successfully solved through SLS methods by performing first a propositionalization and then applying a SAT solver.

Some of the currently best performing SLS algorithms for MAX-SAT are tabu search (TS) algorithms, dynamic local search and ILS. Here, we will describe a combination of a variant of TS with ILS to build a hybrid SAT solver that we will use for inference and learning in SMLNs.

4.5.1 Iterated Local Search and Robust Tabu Search

Many widely known and high-performance local search algorithms make use of randomized choice in generating or selecting candidate solutions for a given combinatorial problem instance. These algorithms are called SLS algorithms [11] and represent one of the most successful and widely used approaches for solving hard combinatorial problem. Many "simple" SLS methods come from other search methods by just randomizing the selection of the candidates during search, such as randomized iterative improvement (RII) and uniformed random walk. Many other SLS methods combine "simple" SLS methods to exploit the abilities of each of these during search. These are known as hybrid SLS methods [11]. ILS is one of these metaheuristics because it can be easily combined with other SLS methods.

One of the simplest and most intuitive ideas for addressing the fundamental issue of escaping local optima is to use two types of SLS steps: one for reaching local optima as efficiently as possible, and the other for effectively escaping local optima.

ILS methods [11, 13] exploit this key idea and essentially use two types of search steps alternatingly to perform a walk in the space of local optima with respect to the given evaluation function. The algorithm works as follows: the search process starts from a randomly selected element of the search space. From this initial candidate solution, a locally optimal solution is obtained by applying a subsidiary local search procedure. Then each iteration step of the algorithm consists of three major substeps: first, a perturbation method is applied to the current candidate solution s; this yields a modified candidate solution s' from which in the next step a subsidiary local search is performed until a local optimum s'' is obtained. In the third step, an acceptance criterion is used to decide from which of the two local optima s or s' the search process is continued. The algorithm can terminate after some steps have not produced improvement or simply after a certain number of steps. The choice of the components of the ILS has a great impact on the performance of the algorithm.

RoTS [27] is a special case of tabu search. In each search step, the RoTS algorithm for MAX-SAT flips a non-tabu variable that achieves a maximal improvement in the total weight of the unsatisfied clauses (the size of this improvement is also called score) and declares it tabu for the next tt steps. The parameter tt is called the tabu tenure. An exception to this tabu rule is made if a more recently flipped variable achieves an improvement over the best solution seen so far (this mechanism is called aspiration). Furthermore, whenever a variable has not been flipped within a certain number of search steps (we use $10n$, n being the number of variables), it is forced to be flipped. This implements a form of long-term memory and helps prevent stagnation of the search process. The tabu status of variables is determined by comparing the number of search steps that have been performed since the most recent flip of a given variable with the current tabu tenure.

4.5.2 Iterated Robust Tabu Search

The original version of IRoTS for MAX-SAT was proposed in [26]. Algorithm 1 starts by independently (with equal probability) initializing the truth values of the atoms. Then it performs a local search to efficiently reach a local optimum CL_S using RoTS. At this point, a perturbation method based again on RoTS is applied leading to the neighbor CL'_C of CL_S and then again a local search based on RoTS is applied to CL'_C to reach another local optimum CL'_S. The *accept* function decides whether the search must continue from the previous local optimum or from the last found local optimum CL'_S. (*accept* can perform random walk or iterative improvement in the space of local optima).

Careful choice of the various components of Algorithm 1 is important to achieve high performance. For the tabu tenure, we refer to the parameters used in [26] which have proven to be highly performant across many domains. At the beginning of each local search and perturbation phase, all variables are declared non-tabu. The clause perturbation operator (flipping the atoms truth value) has the goal to jump in a different region of the search space where search should start with the next

Algorithm 1 Iterated Robust Tabu Search

```
Input: C: set of weighted clauses in CNF, BestScore: current best score)
CL$_C$ = Random initialization of truth values for atoms in C;
CL$_S$ = $LocalSearch_{RoTS}$(CL$_S$);
BestAssignment = CL$_S$;
BestScore = Score(CL$_S$);
repeat
  CL'$_C$ = $Perturb_{RoTS}$(BestAssignment);
  CL'$_S$ = $LocalSearch_{RoTS}$(CL'$_C$);
  if Score(CL'$_S$) $\geq$ BestScore then
    BestScore = Score(CL'$_S$)
  end if
  BestAssignment = accept(BestAssignment,CL'$_S$);
until two consecutive steps have not produced improvement
Return BestAssignment
```

iteration. There can be strong or weak perturbations which means that if the jump in the search space is near to the current local optimum the subsidiary local search procedure $LocalSearch_{RoTS}$ may fall again in the same local optimum and enter regions with the same value of the objective function called *plateau*, but if the jump is too far, $LocalSearch_{RoTS}$ may take too many steps to reach another good solution. In our algorithm, we use a fixed number of RoTS steps $9n/10$ with tabu tenure $n/2$ where n is the number of atoms (in future work, we intend to dynamically adapt the nature of the perturbation). Regarding the procedure $LocalSearch_{RoTS}$, it performs RoTS steps until no improvement is achieved for $n^2/4$ steps with a tabu tenure $n/10 + 4$. The *accept* function always accepts the best solution found so far. The difference between our algorithm and that in [26] is that we do not dynamically adapt the tabu tenure and do not use a probabilistic choice in *accept*.

4.5.3 MAP Inference Using IRoTS

MAP inference in MNs means finding the most likely state of a set of output variables given the state of the input variables. This problem is NP-hard. For discriminative training, the voted perceptron is a special case in which tractable inference is possible using the Viterbi algorithm [4]. In [25], the voted perceptron was generalized to MLNs by replacing the Viterbi algorithm with a weighted SAT solver. This algorithm is gradient descent, and computing the gradient of the conditional log-likelihood (CLL) requires the computation of the number of true groundings for each clause. This can be performed by finding the MAP state which can be computed by dynamic programming methods. Since for MLNs, the MAP state is the state that maximizes the sum of the weights of the satisfied ground clauses, this state can be efficiently found using a weighted MAX-SAT solver. The authors in [25] use the MaxWalkSat solver [24]. In this chapter, we propose to use IRoTS as a MAX-SAT solver and show how this algorithm can be applied not only to MLNs but also to the proposed extension of SMLNs. IRoTS is one the best weighted MAX-SAT solvers and MAP inference in SMLNs can benefit from it.

Given a SMLN in the form of clauses based on *temporal predicates*, including a time predicate and a set of evidence atoms, the KB to be used as input for IRoTS is formed by constructing all groundings of clauses in the SMLNs involving query atoms. Then the evidence atoms are replaced by their true values followed by simplification. Once the SMLN has been propositionalized, it is a natural input to IRoTS.

4.6 Markov Chain Iterated Robust Tabu Search

In this section, we describe how IRoTS can be combined with Markov Chain Monte Carlo (MCMC) to uniformly sample from the space of satisfying assignments of a clause. We show how the proposed algorithm MC-IRoTS can be used for inference and learning in SMLNs.

4.6.1 Conditional Inference Through MC-IRoTS

Conditional inference in graphical models involves computing the distribution of the query variables given the evidence and it has been shown to be #P-complete [29]. The most widely used approach to approximate inference is using MCMC methods and in particular Gibbs sampling. One of the problems that arises in real-world applications is that an inference method must be able to handle probabilistic and deterministic dependencies that might hold in the domain. MCMC methods are suitable for handling probabilistic dependencies but give poor results when deterministic or near deterministic dependencies characterize a certain domain. On the other hand, logical ones such as satisfiability testing cannot be applied to probabilistic dependencies. One approach to deal with both kinds of dependencies is that of [21] where the authors use SampleSAT [28] in a MCMC algorithm to uniformly sample from the set of satisfying solutions. As pointed out in [28], SAT solvers find solutions very fast but they may sample highly non-uniformly. On the other hand, MCMC methods may take exponential time, in terms of problem size, to reach the stationary distribution. For this reason, the authors in [28] proposed to use a hybrid strategy by combining random walk steps with MCMC steps, and in particular with Metropolis transitions. This permits to efficiently jump between isolated or near-isolated regions of non-zero probability, while preserving detailed balance.

We use the same approach as the authors did in [21], but instead of SampleSAT, for MC-IRoTS we propose to use SampleIRoTS, which performs with probability p a RoTS step and with probability $1 - p$ a simulated annealing (SA) step. We used fixed temperature annealing (i.e., Metropolis) moves. The goal is to reach as fast as possible a first solution through IRoTS and then exploit the ability of SA to explore a cluster of solutions. A cluster of solutions is usually a set of connected solutions, so that any two solutions within the cluster can be connected through a series of flips without leaving the cluster. In many domains of interest, solutions exist in clusters and it is highly useful to explore such clusters without leaving them. SA has good properties in exploring a connected space; therefore, it samples near-uniformly and often explores all the neighboring solutions.

Through MC-IRoTS, we can perform conditional inference given evidence to compute probabilities for query predicates. These probabilities can be used to make predictions from the model.

4.6.2 Discriminative Learning by Sampling with MC-IRoTS

Discriminative approaches to weight learning try to optimize the CLL. Preconditioned scaled conjugate gradient (PSCG) is the state-of-the-art discriminative training algorithm for MLN and it was shown in [15] to outperform the voted perceptron. PSCG is a *conjugate gradient* method that uses samples from MC-SAT to approximate the Hessian for MLNs instead of the line search to choose a step size. This approach is also known as *scaled conjugate gradient* and was originally proposed in [20] for training neural networks. PSCG, in each iteration, takes a step in the diagonalized Newton direction (for details, see [15]). Here, we propose to use MC-IRoTS to sample for approximating the Hessian for SMLNs. The goal is to use samples from MC-IRoTS that can serve as good estimates for computing the Hessian.

4.7 Modeling Protein Sequences in SMLNs

In this section, we describe how sequences of protein secondary structure can be modeled in SMLNs, how to learn model parameters from the data, and how to make predictions from the model.

4.7.1 Model Construction and Weight Learning

The approach we follow is quite simple: we write a few formulas that represent the structure of the domain and then from the training sequences we learn the weights of these formulas.

The dataset we refer to is that used in [10]. The data consist of logical sequences of the secondary structure of protein domains:

$beginSequence.$
$strand('SB', null, medium).$

$strand('SB', plus, medium).$
$helix(right, alpha, long).$
...
...
$strand('SB', plus, medium).$
$helix(right, alpha, medium).$
$strand('SB', plus, short).$
endSequence.

A simple SMLN that can be used to model these data can be the following:

//predicates
$Helix(wing, typeh, length, time)$
$Strand(types, pm, length, time)$
$Succ(time, time)$
//rules

$$Helix(Right, +ty1, +l1, +t1) \wedge Succ(t2, t1) \wedge Helix(Right, +ty2, +l2, t2)$$

$$Helix(Right, +ty1, +l1, +t1) \wedge Succ(t2, t1) \wedge Strand(typ1, +p1, +le1, t2)$$

$$Strand(typ1, +p1, +le1, +t1) \wedge Succ(t2, t1) \wedge Helix(Right, +ty2, +l2, t2)$$

$$Strand(typ1, +p1, +le1, +t1) \wedge Succ(t2, t1) \wedge Strand(typ1, +p2, +le2, t2)$$

$$Strand(typ1, +p1, +le1, +t1) \wedge Succ(t2, t1) \wedge Strand(typ2, +p2, +le2, t2)$$

The first three expressions represent the *temporal predicates* (Helix and Strand) and the *time predicate* Succ. The rules express temporal relations between Helix and Strand. The first rule expresses that a helix is followed by another helix, the second rule states that helix is followed by a strand and so on. The last two rules differ in that one states that strand is followed by another strand of the same type, while the other states that two strands of different types appear in the sequence one following the other. The $+$ operator is used to express that the argument should be grounded, so that a weight is learned for each grounded formula. If multiple variables are preceded by $+$, a weight is learned for each combination of their values.

It must be noted that this model is not the only one. We have stated some regularities that in general are true but we would like to learn weights that can express how strong each rule it is. A reasonable model would also be that of learning a rule for each grounding of the argument that expresses the type of strand, i.e., $Strand(+typ1, p1, le1, t1)$. We plan to experiment this in the future.

4.7.2 Making Predictions from the Learned SMLNs

There are several ways in which a SMLNs can be used to make predictions regarding sequences. One way is to predict the sequence with highest probability (this is performed in other models by the Viterbi algorithm). In SMLNs, this means performing MAP inference given the evidence. For example, given a certain sequence and a query predicate (helix or strand), we can perform MAP inference through IRoTS and return the positive query atoms. Then we can count how many of the returned positive query atoms are true in the sequence and consider these atoms *correctly classified*. If there are several models (one for each protein fold), a majority vote can be used to assign the sequence, i.e., the sequence belongs to the model that gives the highest number of *correctly classified* atoms for that sequence.

Another way to get a classifier is to get the probability of query atoms given evidence through MC-IRoTS. In this case, CLL (conditional log-likelihood averaged over all the groundings of the query predicate where predicted probability p is summed for positive atoms and $1 - p$ for negative ones) can be used to assign the sequence to the fold that produces the highest CLL.

4.8 Experiments

The goal of the experiments is to show that SMLNs are suitable for modeling and learning with sequences of logical atoms and that reasonable predictions can be made based on this model.

4.8.1 Dataset and Systems

We implemented the algorithms IRoTS and MC-IRoTS as extensions of the alchemy system [18] and used the implementation of PSCG in this package to learn weights for the SMLN.

The protein fold classification task is to predict one of the five most populated SCOP folds of alpha and beta proteins (a/b). We will use for our experiments only two folds from the dataset in [10] with a subset of the whole sequences. (Our goal here is to show how SMLNs can be used for sequence modeling/learning and not to boost performance or compare with other methods. For this reason we did not optimize any parameters for the learning and inference algorithms). We randomly extracted 80 sequences (totally 160) from each of the folds TIM beta/alpha-barrel (c1) and NAD(P)-binding Rossmann-fold (c2). We divided this set in the training set (60 sequences) and test set (20 sequences). The classification problem is a multiclass one. We will learn for each fold a model based on the SMLN presented in the previous section and will test both models on the 40 test sequences (thus for each model, we have 20 positive and 20 negative testing sequences).

In order to learn weights for the SMLN presented in the previous section we used PSCG with MC-IRoTS as a sampler. We performed two weight learning experiments, one for each protein fold giving in input the same SMLN. We used 100 iterations for PSCG with 50 samples for MC-IRoTS and the lazy version of inference in alchemy [22].

4.8.2 Results

After learning an SMLN on each training set of 60 sequences, we performed MAP inference through IRoTS with both learned models on the two testing sets. For each sequence, every model produced positive query atoms, and we checked the number of predicted positive atoms that were true in the sequence. The sequence was assigned to the model that inferred the highest number of correct predictions. From the 40 test sequences, 31 sequences were correctly assigned to the belonging fold, four sequences were predicted equally by both models and five sequences were classified incorrectly. For the equally predicted sequences, we decided to use MC-IRoTS to perform inference over these sequence to get CLL for each one. Surprisingly, using the probability for each atom in the sequence, all these cases were correctly classified; thus we achieved a classification rate of 35 out of 40.

These preliminary results are promising, showing that the modeling power of the described approach is suitable for the problem of protein fold recognition. Moreover, the model can be used to make predictions which seem reasonably good. However, further exploration of alternative experimental evaluation could help to perform extensive experiments and compare with state-of-the-art methods such that in [10].

4.9 Conclusions

MLNs are a powerful representation that combines first-order logic and probability by attaching weights to first-order formulas and viewing these as templates for features of MNs. In this chapter we have described SMLNs, a simple extension of MLN that is able to deal with sequences of logical atoms. We also propose iterated robust tabu search (IRoTS) for MAP inference in SMLNs and Markov Chain-IRoTS (MC-IRoTS) for conditional inference in SMLNs. We show how MC-IRoTS can also be used for discriminative weight learning in SMLNs. As application domain, we have described how sequences of protein secondary structure can be modeled in SMLNs and have shown through some preliminary experiments the promise of our approach.

Regarding SMLNs in general, we would like to apply this simple extension of MLNs to more complex domains where stochastic relational problems must be handled. Natural application domains for SMLNs are areas such as Systems Biology and Gene Regulatory Networks where networks involved in stochastic processes need to be modeled.

Acknowledgements This work was partially funded by project DM19410 – Laboratorio di Bioinformatica per la Biodiversità Molecolare. We thank Kristian Kersting for providing us the dataset on protein sequences.

References

1. Besag, J. Statistical Analysis of Non-lattice Data. *Statistician*, 24:179–195, 1975
2. Biba, M., Ferilli, S., and Esposito, F. Structure Learning of Markov Logic Networks through Iterated Local Search. In Proc. 18th European Conference on Artificial Intelligence, 2008
3. Biba, M., Ferilli, S., and Esposito, F. Discriminative Structure Learning of Markov Logic Networks. In Proceedings of 18th International Conference on Inductive Logic Programming (ILP 2008), LNCS 5194, pp. 59–76. Berlin: Springer, 2008
4. Collins, M. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In Proc. of the 2002 Conference on Empirical Methods in Natural Language Processing, pp. 1–8. Philadelphia, PA: ACL, 2002
5. Della Pietra, S., Pietra, V. D., and Laferty, J. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:380–392, 1997
6. Dick, U. and Kersting, K. Fisher Kernels for Relational Data. In J. Fuernkranz, T. Scheffer, M. Spiliopoulou, editors, Proc. 17th ECML, pp. 114–125, 2006
7. Dietterich, T., Ashenfelter, A., and Bulatov, Y. Training conditional random fields via gradient tree boosting. ICML 2004
8. Genesereth, M. R. and Nilsson, N. J. *Logical foundations of artificial intelligence*. San Mateo, CA: Morgan Kaufmann, 1987
9. Getoor, L. and Taskar, B. *Introduction to statistical relational learning*. MA: MIT, 2007
10. Gutmann, B. and Kersting, K. TildeCRF: Conditional Random Fields for Logical Sequences. In J. Fuernkranz, T. Scheffer, M. Spiliopoulou, editors, Proc. of the 17th ECML, pp. 174–185, 2006
11. Hoos, H. H. and Stutzle, T. *Stochastic local search: Foundations and applications*. San Francisco: Morgan Kaufmann, 2005
12. Laferty, J., McCallum, A., and Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proc. 18th Int'l Conf. on Machine Learning, pp. 282–289, 2001
13. Loureno, H. R., Martin, O., and Stutzle, T. Iterated local search. In *Handbook of metaheuristics*, pp. 321–353. Dordrecht: Kluwer, 2002
14. Mihalkova, L. and Mooney, R. J. Bottom-Up Learning of Markov Logic Network Structure. In Proc. 24th Int'l Conf. on Machine Learning, pp. 625–632, 2007
15. Lowd, D. and Domingos, P. Efficient Weight Learning for Markov Logic Networks. In Proc. of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 200–211, 2007
16. Kersting, K., De Raedt, L., and Raiko, T. Logial Hidden Markov Models. *Journal of Artificial Intelligence Research (JAIR)*, 25:425–456, 2006
17. Kok, S. and Domingos, P. Learning the Structure of Markov Logic Networks. In Proc. 22nd Int'l Conf. on Machine Learning, pp. 441–448, 2005
18. Kok, S., Singla, P., Richardson, M., and Domingos, P. The alchemy system for statistical relational ai (Technical Report). Department of Computer Science and Engineering, University of Washington, Seattle, WA, 2005. http://alchemy.cs.washington.edu/
19. Manna Z. and Pnueli, A. *The temporal logic of reactive and concurrent systems*. Berlin: Springer, 1992
20. Moller, M. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, 1993

21. Poon, H. and Domingos, P. Sound and Efficient Inference with Probabilistic and Deterministic Dependencies. In Proc. 21st Nat'l Conf. on Artificial Intelligence, pp. 458–463. Chicago, IL: AAAI Press, 2006
22. Poon, H., Domingos, P., and Sumner, M. A General Method for Reducing the Complexity of Relational Inference and its Application to MCMC. In Proc. 23rd Nat'l Conf. on Artificial Intelligence. Chicago, IL: AAAI Press, 2008
23. Richardson, M. and Domingos, P. Markov logic networks. *Machine Learning*, 62:107–236, 2006
24. Selman, B., Kautz, H., and Cohen, B. Local search strategies for satisfiability testing. In D. S. Johnson and M. A. Trick, editors, Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, American Mathematical Society, pp. 521–532, 1996
25. Singla, P. and Domingos, P. Discriminative Training of Markov Logic Networks. In Proc. 20th Nat'l Conf. on Artificial Intelligence, pp. 868–873. Chicago, IL: AAAI Press, 2005
26. Smyth, K., Hoos, H., and Stutzle, T. Iterated Robust Tabu Search for MAX-SAT. Canadian Conference on AI, pp. 129–144, 2003
27. Taillard, E. D. Robust taboo search for the quadratic assignment problem. *Parallel Computing*, 17:443–455, 1991
28. Wei, W., Erenrich, J., and Selman, B. Towards Efficient Sampling: Exploiting Random Walk Strategies. National Conference on Artificial Intelligence, AAAI Press, 2004
29. Roth, D. On the hardness of approximate reasoning. *Artificial Intelligence*, 82:273–302, 1996
30. Liu, D. C. and Nocedal, J. On the limited memory BFGS method for large scale optimization, *Mathematical Programming*, 45:503–528, 1989
31. M. Moller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, 1993

Chapter 5
Mining Spatial Association Rules for Composite Motif Discovery

Michelangelo Ceci, Corrado Loglisci, Eliana Salvemini, Domenica D'Elia, and Donato Malerba

Abstract Motif discovery in biological sequences is an important field in bioinformatics. Most of the scientific research focuses on the de novo discovery of single motifs, but biological activities are typically co-regulated by several factors and this feature is properly reflected by higher order structures, called composite motifs, or cis-regulatory modules or simply modules. A module is a set of motifs, constrained both in number and location, which is statistically overrepresented and hence may be indicative of a biological function. Several methods have been studied for the de novo discovery of modules. We propose an alternative approach based on the discovery of rules that define strong spatial associations between single motifs and suggest the structure of a module. Single motifs involved in the mined rules might be either de novo discovered by motif discovery algorithms or taken from databases of single motifs. Rules are expressed in a first-order logic formalism and are mined by means of an inductive logic programming system. We also propose computational solutions to two issues: the hard discretization of numerical inter-motif distances and the choice of a minimum support threshold. All methods have been implemented and integrated in a tool designed to support biologists in the discovery and characterization of composite motifs. A case study is reported in order to show the potential of the tool.

5.1 Introduction

In biological sequence analysis, a *motif* is a nucleotide or amino-acid sequence pattern which appears in a set of sequences (DNA, RNA or protein) with much higher frequency than would be expected by chance. This statistical overrepresentation is

D. Malerba (✉)
Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro," via Orabona 4,
70126 Bari, Italy
e-mail: malerba@di.uniba.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_5,

expected to be indicative of an associated biological function. Examples of motifs include DNA- and RNA-binding sites for regulatory proteins, protein domains and protein epitopes.

DNA and RNA motifs are key to deciphering the language of gene regulatory mechanisms and, in particular, to fully understand how gene expression is regulated in time and space. For this reason, de novo (or ab initio) motif discovery, i.e. identifying motif sites (signals) in a given set of unaligned biological sequences, has attracted the attention of many biologists. However, they are also difficult to identify, since motifs often produce weak signals buried in genomic noise (i.e. the background sequence) [8]. This problem is known to be NP-hard [22], thus it is also an interesting arena for computer scientists.

Most of the motif discovery tools reported in the literature are designed to discover single motifs. However, in many (if not most) cases, biological activities are co-regulated by several factors. For instance, transcription factor-binding sites (TFBSs) on DNA are often organized in functional groups called *composite motifs* or *cis-regulatory modules* (CRM) or simply modules. These modules may have a biologically important structure that constrains both the number and relative position of the constituent motifs [34].

One example, among many that could be cited, is ETS-CBF, a cis-regulatory module constituted by three single motifs, μA, μB and CBF (core-binding factor). Both μA and μB are binding sites for two transcription factors belonging to the ETS proteins family, Ets-1 and PU.1, respectively. CBF is a protein that is implicated in the activation of several T and myeloid cell-specific promoters and enhancers. Enhancers are cis-regulatory sequences which control the efficiency of gene transcription from an adjacent promoter. ETS-CBF is a common composite motif of enhancers implicated in the regulation of antigen receptor genes in mouse and human. A comparative study of the tripartite domain of the murine immunoglobulin μ heavy-chain (IgH) enhancer and its homologous in human has demonstrated that in both species the activity of the gene enhancer is strictly dependent on ETS-CBF [12].

Therefore, it is of great interest to discover not only single motifs but also the higher order structure into which motifs are organized, i.e. the modules. This problem is also known as *composite* [38] or *structured* [32] *motif* discovery.

Over the past few years, a plethora of single motif discovery tools have been reported in the literature (see the book by Robin et al. [36]). They differ in three aspects:

1. The representation of a pattern that constitutes a single motif,
2. The definition of overrepresentation of a motif pattern and
3. The search strategy applied to pattern finding.

A single motif can be represented either by a consensus sequence, which contains the most frequent nucleotide in each position of the observed signals, or by a position weight matrix (PWM), which assigns a different probability to each possible letter at each position in the motif [46].

Both consensus sequences and PWMs are derived by the multiple alignments of all the known recognition sites for a given regulatory factor and represent the specificity of a regulatory factor for its own recognition site. They refer to a sequence that matches all the sequences of the aligned recognition sites very closely, but not necessarily exactly. In a consensus sequence, this concept is expressed by notations that indicate which positions of the consensus sequence are always occupied by the same nucleotide (exact match) and which one can vary and how (allowed mismatch), without affecting the functionality of the motif. Considering the example DNA consensus sequence T[CT]NG{A}A, it has to be read in the following way: the first, fourth and sixth position in the consensus are always occupied by T, G and A, where T stands for thymidine, G for guanine and A for adenine; no mismatches are allowed in these positions. The second nucleotide in the sequence can be a cytosine (C) or alternatively a T. This mismatch does not affect the effectiveness of the recognition signal. The third position of the consensus can be occupied by any of the four nucleotide bases (A, T, C, G). At the fifth position any base can be present except A.

A PWM of a DNA motif has one row for each nucleotide base (A, T, C, G) and one column for each position in the pattern. This way, there is a matrix element for all possible basis at every position. The score of any particular motif in a sequence of DNA is the sum of the matrix values for that motif's sequence. This score is the same of the consensus only when the motif perfectly matches the consensus. Any sequence motif that differs from the consensus in some positions will have a lower score depending on the number and type of nucleotide mismatches.

In contrast to these sequence patterns, spatial patterns have also been investigated [19], where spatial relationships (e.g. adjacency and parallelism) and shapes (e.g., α-helices in protein motifs) can be represented.

The overrepresentation of motif patterns has been defined in several ways. In some motif-discovery algorithms, a score is defined for each pattern (e.g., p-value [47] or z-score [43]), and the observed motif scores are compared with expected scores from a background model. In other algorithms, two separate values are computed when evaluating motifs, one concerning the support, or coverage, of a motif, and the other concerning the unexpectedness of a motif [35]. A third approach is to use a measure of information content [25] of discovered patterns.

Search strategies can be categorized as enumerative (or pattern-driven) and heuristic (or sequence-driven). The former enumerate all possible motifs in a given solution space (defined by a template pattern) and test each for significance, while the latter try to build a motif model by varying some model parameters such that a matching score with sequence data is maximized. In general, enumerative algorithms find optimal solutions for discrete representations of relatively short motifs, but do not scale well to larger motifs and continuous models. TEIRESIAS [35] is more sophisticated in using information about the relative occurrences of substrings; therefore, it can be used to discover discrete representations of longer motifs. Among the heuristic-based approaches, the most common is the expectation-maximization (EM) [5], which is a deterministic local search algorithm. EM may converge very fast, but the optimality of the returned point strongly depends on the starting point (seed). For this reason, it is used in combination with some randomization techniques in order to escape from a poor local optimum even if the chosen seed is bad [6].

Algorithms for the de novo discovery of modules, together with the parameters of their constituent motifs [14, 41, 52], are more recent. These algorithms, which exploit some form of spatial information (e.g., spatial correlation) on constituent motifs to identify a module, are considered particularly promising since they may offer both improved performance over conventional discovery algorithms of single motifs and insight into the mechanism of regulation directed by the constituent motifs [26]. However, in order to restrict the search space, they make some assumptions which limit their flexibility in handling variations of either the number or length of the constituent motifs or the spacing between them. For instance, the hierarchical mixture (HMx) model of CISMODULE [52] requires the specification of both the length of the module and the total number of constituent motif types. Moreover, CISMODULE does not capture the order or precise spacing of multiple TFBSs in a module. Segal and Sharan [41] propose a method for the de novo discovery of modules consisting of a combination of single motifs which are spatially close to each other. Despite the flexibility of their method in handling modules, they assume that a training set (with positive and negative examples of transcriptional regulation) is available in order to learn a discriminative model of modules. The method EMC module proposed by Gupta and Liu [14] assumes a geometrical probability distribution on the distance between TFBSs.

Although a recent study [18] has shown a significant improvement in prediction success when modules are considered instead of isolated motifs, it is largely believed that without some strong form of inductive bias,[1] methods for de novo module discovery may have performance close to random. For this reason, another line of module discovery methods has been investigated (e.g., Cister [13], Module-Searcher [1], MScan [20], Compo [39]), which takes a list of single motifs as input along with the sequence data in which the modules should be found. Single motifs are taken from motif databases, such as TRANSFAC [15] and JASPAR [37], and the challenges concern discovering which of them are involved in the module, defining the sequence of single motifs in the module and possibly discovering the inter-motif distances.[2]

Module discovery methods can be categorized according to the type of framework, either discrete (e.g., CREME [42]) or probabilistic (e.g., Logos [51]), adopted to model modules. In a discrete framework, all constituent motifs must appear in a module instance. This simplifies inference and interpretation of modules, and often allows exhaustive search of optimal constituent motifs in a sequence window

[1] The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered. It forms the rationale for learning since without it no generalization is possible [29].

[2] The distance is typically evaluated as the number of nucleotides which separate two consecutive single motifs. More sophisticated distance measures might be used in future works if significant progress is made in the prediction of DNA folding.

of a given length. Conversely, a probabilistic framework is more expressive, since it relaxes the hard constraints of discrete frameworks and associates each module with a score which is a combination (e.g., the sum) of motifs and distance scores. Issues of probabilistic frameworks are local optima and interpretability of results.

A recent assessment of eight published methods for module discovery [21] has shown that no single method performed consistently better than others in all situations and that there are still advances to be made in computational module discovery. In this chapter, we propose an innovative approach to module discovery, which can be a useful supplement or alternative to other well-known approaches. The idea is to mine rules which define "strong" spatial associations between single motifs [27]. Single motifs might either be de novo discovered by traditional discovery algorithms or taken from databases of known motifs.

The spatial relationships considered in this work are the order of motifs along the DNA sequence and the inter-motif distance between each consecutive couple of motifs, although the mining method proposed to generate spatial association rules has no limitation on both the number and the nature of spatial relationships. The association rule mining method is based on an inductive logic programming (ILP) [31] formulation according to which both data and discovered patterns are represented in a first-order logic formalism. This formulation also facilitates the accommodation of diverse sources of domain (or background) knowledge which are expressed in a declarative way. Indeed, ILP is particularly well suited to bioinformatics tasks due to its ability both to take into account background knowledge and to work directly with structured data [30]. This is confirmed by some notable success in molecular biology applications, such as predicting carcinogenesis [44, 45].

The proposed approach is based on a discrete framework, which presents several advantages, the most relevant being the straightforward interpretation of rules, but also some disadvantages, such as the hard discretization of numerical inter-motif distances or the choice of a minimum support threshold. To overcome these issues, some computational solutions have been developed and tested.

The specific features of this approach are:

- An original perspective of module discovery as a spatial association rule mining task;
- A logic-based approach where background knowledge can be expressed in a declarative way;
- A procedure for the automated selection of some parameters which are difficult to properly set;
- Some computational solutions to overcome the discretization issues of discrete approaches.

These features provide our module discovery tool several advantages with respect to competitive approaches. First, spatial association rules, which take the form of $A \Rightarrow C$, provide insight both into the support of the module (represented by $A \wedge C$) and into the confidence of possible predictions of C given A. Predictions may equally concern both properties of motifs (e.g., its type) and spatial relationships (e.g., the inter-motif distance). Second, the declarative knowledge representation facilitates the development and debugging of background knowledge in collaboration with a domain expert. Moreover, knowledge expressed in a declarative way is re-usable across different tasks and domains, thus easing the burden of the knowledge engineering effort. Third, the resort to first-order (or relational) logic facilitates the representation of input sequences, whose structure can be arbitrarily complex, and increases the explanatory power of discovered patterns, which are relatively easy to interpret for domain experts. Fourth, computational solutions devised for both the problem of selecting a minimum support threshold and the problem of discretizing numerical data fulfill the twofold goal of improving the quality of results and designing tools for the actual end-users, namely biologists.

Further significant advantages are:

- No prior assumption is necessary either on the constituent motifs of a module or on their spatial distribution;
- Specific information on the bases occurring between two consecutive motifs is not required.

This work also extends our previous study [48], where frequent patterns are generated by means of the algorithm GSP [3]. The extension aims to: (1) find association rules, which convey additional information with respect to frequent patterns; (2) discover more significant inter-motif distances by means of a new discretization algorithm which does not require input parameters; (3) automatically select the best minimum support threshold; (4) filter redundant rules; (5) investigate a new application of an ILP algorithm to a challenging bioinformatics task.

The chapter is organized as follows. Section 5.2 presents a formalization of the problem, which is decomposed into two subproblems: (1) mining frequent sets of motifs, and (2) mining spatial association rules. Input and output of each step of the proposed approach are also reported. Section 5.3 describes the method for spatial association rule mining. Section 5.4 presents the solution to some methodological and architectural problems which affect the implementation of a module discovery tool effectively usable by biologists. Section 5.5 is devoted to a case study, which shows the application of the developed system. Finally, conclusions are drawn.

5.2 Mining Spatial Association Rules from Sequences

Before proceeding to a formalization of the problem, we first introduce some general notions on association rules.

Association rules are a class of patterns that describe regularities or co-occurrence relationships in a set T of homogeneous data structures (e.g., sets, sequences and so on) [2]. Formally, an association rule R is expressed in the form of $A \Rightarrow C$, where A (the *antecedent*) and C (the *consequent*) are disjoint conditions on properties of data structures (e.g., the presence of an item in a set). The meaning of an association rule is quite intuitive: if a data structure satisfies A, then it is likely to satisfy C. To quantify this likelihood, two statistical parameters are usually computed, namely *support* and *confidence*. The former, denoted as $sup(R, T)$, estimates the probability $P(A \wedge C)$ by means of the percentage of data structures in T satisfying both A and C. The latter, denoted as $conf(R, T)$, estimates the probability $P(C|A)$ by means of the percentage of data structures which satisfy condition C, out of those which satisfy condition A. The task of association rule mining consists in discovering all rules whose support and confidence values exceed two respective minimum thresholds. When data structures describe spatial objects together with their spatial relationships, mined association rules are called *spatial*, since conditions in either the antecedent or the consequent of a rule express some form of spatial constraint.

We now give a formal statement of the module discovery problem, which is decomposed into two subproblems as follows:

1. *Given:* A set M of single motifs, a set T of sequences with annotations about type and position of motifs in M and a minimum value τ_{min},
   Find: The collection $\mathcal{S}$ of all the sets S_1, $S_2, \ldots, S_n$ of single motifs such that, for each S_i, at least τ_{min} sequences in T contain all motifs in S_i.
2. *Given:* A set $S \in \mathcal{S}$ and two thresholds σ_{min} and κ_{min},
   Find: Spatial association rules involving motifs in S, such that their support and confidence are greater than σ_{min} and κ_{min}, respectively.

Single motifs in M can be either discovered de novo or taken from a single motif database. Each $S_i \in \mathcal{S}$ is called *motif set*. The *support set* of S_i is the subset T_{S_i} of sequences in T such that each sequence in T_{S_i} contains at least one occurrence of each motif in S_i. According to the statement of subproblem (1) $|T_{S_i}| \geq \tau_{min}$. T_{S_i} is used to evaluate both support and confidence of spatial association rules mentioned in subproblem (2).

The proposed approach is two-stepped since it reflects this problem decomposition. In the first step, motif sets which are *frequent*, i.e., have a support greater than τ_{min}, are extracted from sequences annotated with predictions for known single motifs. Only information about the occurrence of motifs is considered, while spatial distribution of motifs is ignored. This step has a manifold purpose: (1) enabling biologists to guide deeper analysis only for sets of motifs which are deemed potentially interesting; (2) filtering out sequences which do not include those interesting sets of motifs; (3) lowering the computational cost of the second step.

In the second step, sequences that support specific frequent motif sets are abstracted into *sequences of spaced motifs*. A sequence of spaced motifs is defined as an ordered collection of motifs interleaved with inter-motif distances. Each inter-motif distance measures the distance between the last nucleotide of a motif and the first nucleotide of the next motif in the sequence. Spatial association rules are mined from these abstractions. In order to deal with numerical information on the inter-motif distance, a discretization algorithm is applied. The algorithm takes into account the distribution of the examples and does not significantly depend on input parameters as in the case of classical equal width or equal frequency discretization algorithms. Details on both steps are reported below.

5.2.1 *Mining Frequent Motif Sets*

To solve the first sub-problem, we resort to the levelwise breadth-first search [28] in the lattice of motif sets. The search starts from the smallest element, i.e., sets with one motif in M, and proceeds from smaller to larger sets. The frequent sets of i motifs are used to generate candidate sets of $(i + 1)$ motifs. Each candidate set, which is potentially frequent, is evaluated against the set T of sequences, in order to check the actual size of its support set. Then it is pruned if its support is lower than τ_{min}. For instance, given $M = \{x, y, z\}$ and T as in Fig. 5.1a, the set $S = \{x, y\}$ is supported by $T_S = \{t_2, t_3\}$. If $\tau_{min} = 2$, then S is returned together with other frequent motif sets in $\mathcal{S}$.

5.2.2 *Mining Spatial Association Rules*

The sequences in the support set T_S of a frequent motif set S are represented as chains of the form $\langle m_1, d_1, m_2, \ldots, d_{n-1}, m_n\rangle$, where each m_i denotes a single motif ($m_i \in M$), while each d_i, $i = 1, 2, \ldots, n - 1$, denotes the inter-motif distance between m_i and m_{i+1}. Each chain is a sequence of spaced motifs. For instance, sequence t_2 in Fig. 5.1a is represented as $\langle x, 10, y, 92, y\rangle$.

From a biological viewpoint, slight differences in inter-motif distances can be ignored. For this reason, we can group almost equal distances by applying a discretization technique which maps numerical values into a set of closed intervals.

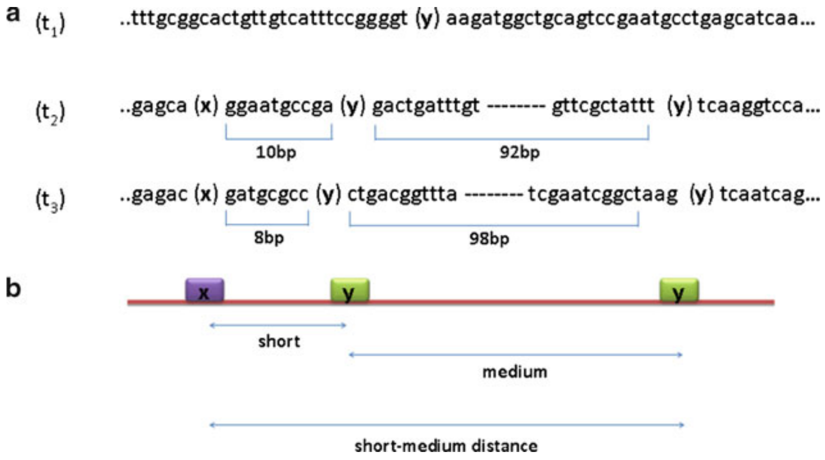

a
(t1)
..tttgcggcactgttgtcatttccggggt (y) aagatggctgcagtccgaatgcctgagcatcaa...
(t2)
..gagca (x) ggaatgccga (y) gactgatttgt -------- gttcgctattt (y) tcaaggtcca...
10bp
92bp
(t3)
..gagac (x) gatgcgcc (y) ctgacggttta -------- tcgaatcggctaag (y) tcaatcag...
8bp
98bp
b
x
y
y
short
medium
short-medium distance

Fig. 5.1 (**a**) Three different annotated sequences (t_1, t_2, t_3) belonging to the set T where motifs x and y have been found. The *grey* semi-boxes underline the nucleotide sequences between two consecutive motifs (inter-motif distance). Inter-motif distances are expressed in base pairs (bp). (**b**) Closed-intervals of inter-motif distances

Therefore, a sequence of spaced motifs can be further abstracted into an ordered collection of motifs interleaved by symbols (e.g., short, medium, and large) representing a range of inter-motif distance. For instance, by considering the closed intervals in Fig. 5.1b, both sequences t_2 and t_3 in Fig. 5.1a are represented by the following sequence of spaced motifs:

$$\langle x, short, y, medium, y\rangle. \tag{5.1}$$

Each sequence of spaced motifs is described in a logic formalism which can be processed by the ILP system SPADA (**S**patial **Pa**ttern **D**iscovery **A**lgorithm) [24] to generate spatial association rules. More precisely, the whole sequence, the constituent motifs and the inter-motif distances are represented by distinct constant symbols.[3] Some predicate symbols are introduced in order to express both properties and relationships. They are:

- *sequence(t)*: t is a sequence of spaced motifs;
- *part_of(t,m)*: The sequence t contains an occurrence m of single motif;
- *is_a(m,x)*: The occurrence m is a motif x;
- *distance(m_1,m_2,d)*: The distance between the occurrences m_1 and m_2 is d.

A sequence is represented by a set of *Datalog*[4] *ground atoms*, where a Datalog ground atom is an n-ary predicate symbol applied to n constants. For instance, the sequence of spaced motif in (5.1) is described by the following set of Datalog ground atoms:

$$\left\{\begin{array}{c} sequence(t_2), \\ part\_of(t_2,m_1),\ part\_of(t_2,m_2),\ part\_of(t_2,m_3), \\ is\_a(m_1,x),\ is\_a(m_2,y),\ is\_a(m_3,y), \\ distance(m_1,m_2,short),\ distance(m_2,m_3,medium). \end{array}\right\} \tag{5.2}$$

The set of Datalog ground atoms of all sequences is stored in the *extensional* part D_E of a deductive database D. The *intensional* part D_I of the deductive database D includes the definition of the domain knowledge in the form of *Datalog rules*. An example of Datalog rules is the following:

$$\begin{array}{c} short\_medium\_distance(U,V) \leftarrow distance(U,V,short). \\ short\_medium\_distance(U,V) \leftarrow distance(U,V,medium). \end{array} \tag{5.3}$$

They state that two motifs[5] are at a *short_medium_distance* if they are at either *short* or *medium* distance (Fig. 5.1b). Rules in D_I allows additional Datalog

[3] We denote constants as strings of lowercase letters possibly followed by subscripts.

[4] Datalog is a query language for deductive databases [9].

[5] Variables are denoted by uppercase letters possibly followed by subscripts, such as U and V.

ground atoms to be deduced from data stored in D_E. For instance, rules in (5.3) entail the following information from the set of Datalog ground atoms in (5.2):

$$\begin{Bmatrix} \mathit{short\_medium\_distance}(m_1, m_2), \\ \mathit{short\_medium\_distance}(m_2, m_3). \end{Bmatrix} \tag{5.4}$$

SPADA adds these entailed Datalog ground atoms to set (5.2), so that atoms with the predicate *short_medium_distance* can also appear in mined association rules.

Spatial association rules discovered by SPADA take the form $A \Rightarrow C$, where both A and C are conjunctions of *Datalog non-ground atoms*. A Datalog ground atom is an n-ary predicate symbol applied to n terms (either constants or variables), at least one of which is a variable. For each association rule, there is exactly one variable denoting the whole sequence and other variables denoting constituent motifs. An example of a spatial association rule is the following:

$$\begin{aligned} &\mathit{sequence}(T), \mathit{part\_of}(T, M_1), \mathit{is\_a}(M_1, x), \mathit{distance}(M_1, M_2, \mathit{short}), \\ &\quad M_1 \neq M_2 \Rightarrow \mathit{is\_a}(M_2, y) \end{aligned} \tag{5.5}$$

where variable T denotes a sequence, while variables M_1 and M_2 denote two distinct occurrences of single motifs ($M_1 \neq M_2$) of type x and y, respectively. With reference to the sequence described in (5.2), T corresponds to t_2 while the two distinct occurrences of single motifs M_1 and M_2 correspond to m_1 and m_2, respectively. By means of this association rule, it is possible to infer which is the single motif that follows in a short distance a single motif x. The uncertainty of the inference is quantified by the confidence of the association rule.

Details on the association rule discovery algorithm implemented in SPADA are reported in the next section.

5.3 SPADA: Pattern Space and Search Procedure

In SPADA, the set O of spatial objects is partitioned into a set S of *reference* (or target) *objects* and m sets R_k, $1 \leq k \leq m$, of *task-relevant* (or non-target) objects. Reference objects are the main subject of analysis and contribute to the computation of the support of a pattern, while task-relevant objects are related to the reference objects and contribute to accounting for the variation, i.e., they can be involved in a pattern. In the sequence described in (5.2), the constant t_2 denotes a reference object, while the constants m_1, m_2 and m_3 denote three task relevant objects. In this case, there is only one set R_1 of task-relevant objects.

SPADA is the only ILP system which addresses the task of relational frequent pattern discovery by dealing properly with concept hierarchies. Indeed, for each set R_k, a generalization hierarchy H_k is defined together with a function ψ_k, which maps objects in H_k into a set of granularity levels $\{1, \ldots, L\}$. For instance, with

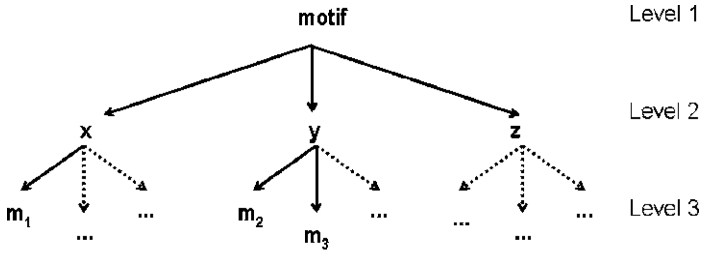


Fig. 5.2 A three-level hierarchy defined on motifs

reference to the sequence described in (5.2), it is possible to define a three-level hierarchy H_1 (Fig. 5.2), where the top level represents a generic single motif, the middle level represents distinct single motifs in M and the lowest level represents specific occurrences of motifs. In this example, the function ψ_1 simply maps the root to 1, x, y, and z to 2 and m_1, m_2 and m_3 to 3.

The set of predicates used in SPADA can be categorized into four classes. The *key predicate* identifies the reference objects in S (e.g., *sequence* is the key predicate in description (5.2)). The *property predicates* are binary predicates which define the value taken by an attribute of an object (e.g., *length* of a motif, not reported in description (5.2)). The *structural predicates* are binary predicates which relate task-relevant objects (e.g., *distance*) as well as reference objects with task-relevant objects (e.g., *part_of*). The *is_a* predicate is a binary *taxonomic* predicate which associates a task-relevant object with a value of some H_k.

The *units of analysis* $D[s]$, one for each reference object $s \in S$, are subsets of ground facts in D_E, defined as follows:

$$D[s] = is\_a(R(s)) \cup D[s|R(s)] \cup \bigcup_{r_i \in R(s)} D[r_i|R(s)], \tag{5.6}$$

where:

- $R(s)$ is the set of task-relevant objects directly or indirectly related to s;
- $is\_a(R(s))$ is the set of *is_a* atoms specified for each $r_i \in R(s)$;
- $D[s|R(s)]$ contains both properties of s and relations between s and some $r_i \in R(s)$;
- $D[r_i|R(s)]$ contains both properties of r_i and relations between r_i and some $r_j \in R(s)$.

This notion of unit of analysis is coherent with the individual-centered representation [7], which has some nice properties, both theoretical (e.g., PAC-learnability [49]) and computational (e.g., smaller hypothesis space and more efficient search). The set of units of analysis is a partitioning of D_E into a number of subsets $D[s]$, each of which includes ground atoms concerning the task-relevant objects (transitively) related to the reference object s. With reference to the sequence described in (5.2), $R(t_2) = \{m_1, m_2, m_3\}$, and $D[t_2]$ coincides with the whole set of ground atoms, including those inferred by means of rules in the intensional part D_I of the deductive database. If several reference objects had been reported in (5.2), $D[t_2]$ would have been a proper subset.

Patterns discovered by SPADA are conjunctions of Datalog non-ground atoms, which can be expressed by means of a set notation. For this reason they are also called *atomsets* [10], by analogy with itemsets introduced for classical association rules. A formal definition of atomset is reported in the following.

Definition 5.1. An *atomset* P is a set of atoms $p_0(t_0^1)$, $p_1(t_1^1, t_1^2)$, $p_2(t_2^1, t_2^2)$, ..., $p_r(t_r^1, t_r^2)$, where p_0 is the key predicate, while p_i, $i = 1, \ldots, r$, is either a structural predicate or a property predicate or an *is_a* predicate.

Terms t_i^j are either constants, which correspond to values of property predicates, or variables, which identify reference objects either in S or in some R_k. Each p_i is a predicate occurring either in D_E (extensionally defined predicate) or in D_I (intensionally defined predicate). Some examples of atomsets are the following:

$$\begin{aligned}
P_1 &\equiv \mathit{sequence}(T), \mathit{part\_of}(T, M_1), \mathit{is\_a}(M_1, x)\\
P_2 &\equiv \mathit{sequence}(T), \mathit{part\_of}(T, M_1), \mathit{is\_a}(M_1, x), \mathit{distance}(M_1, M_2, \mathit{short})\\
P_3 &\equiv \mathit{sequence}(T), \mathit{part\_of}(T, M_1), \mathit{is\_a}(M_1, x), \mathit{distance}(M_1, M_2, \mathit{short}),\\
&\quad\ \mathit{is\_a}(M_2, y)
\end{aligned}$$

where variable T denotes a reference object, while variables M_1 and M_2 denote some task-relevant objects. All variables are implicitly existentially quantified.

Atomsets in the search space explored by SPADA satisfy the *linkedness* [16] property, which means that each variable denoting a task-relevant object in an atomset P defined as in Definition 5.1 must be transitively linked to the reference object t_0^1 by means of structural predicates. For instance, variables M_1 and M_2 in P_1, P_2 and P_3 are transitively linked to T by means of the structural predicates *distance* and *part_of*. Therefore, P_1, P_2 and P_3 satisfy the linkedness property.

Each atomset P is associated with a granularity level l. This means that all taxonomic (*is_a*) atoms in P refer to task-relevant objects, which are mapped by some ψ_k into the same granularity level l. For instance, atomsets P_1, P_2 and P_3 are associated with the granularity level 2 according to the hierarchy H_1 in Fig. 5.2 and the associated function ψ_1. For the same reason, the following atomset:

$$P_4 \equiv \mathit{sequence}(T), \mathit{part\_of}(T, M_1), \mathit{is\_a}(M_1, \mathit{motif})$$

is associated with the granularity level 1.

In multi-level association rule mining, it is possible to define an *ancestor* relation between two atomsets P and P' at different granularity levels.

Definition 5.2. An atomset P at granularity level l is an *ancestor* of an atomset P' at granularity level l', $l < l'$, if P'can be obtained from P by replacing each task-relevant object $h \in H_k$ at granularity level l ($l = \psi_k(h)$) with a task-relevant object h', which is more specific than h in H_k and is mapped into the granularity level l' ($l' = \psi_k(h')$).

For instance, the atomset P_4 defined above is an ancestor of P_1, since P_1 can be obtained from P_4 by replacing *motif* with x.

By associating an atomset P with an existentially quantified conjunctive formula *eqc(P)* obtained by transforming P into a Datalog query, we can now provide a formal definition of the support of P on a deductive database D. We recall that D has an extensional part D_E and an intensional part D_I. Moreover D_E includes several units of analysis $D[s]$ one for each reference object.

Definition 5.3. An atomset P *covers* a unit of analysis $D[s]$ if $D[s] \cup D_I$ logically entails $eqc(P)$ ($D[s] \cup D_I \models eqc(P)$).

Each atomset P is associated with a support, denoted as *sup(P,D)*, which is the percentage of units of analysis in D covered by P. The minimum support for frequent atomsets depends on the granularity level l of task-relevant objects. It is denoted as $\sigma_{min}[l]$ and we assume that $\sigma_{min}[l+1] \leq \sigma_{min}[l]$, $l = 1, 2, \ldots, L$-1.

Definition 5.4. An atomset P at granularity level l with support *sup(P,D)* is *frequent* if $sup(P, D) \geq \sigma_{min}[l]$ and all ancestors of P are frequent at their corresponding levels.

In SPADA, the discovery of frequent atomsets is performed according to both an intra-level and an inter-level search. The intra-level search explores the space of patterns at the same level of granularity. It is based on the level-wise method [28], which performs a breadth-first search of the space, from the most general to the most specific patterns, and prunes portions of the search space which contain only infrequent patterns.

The application of the level-wise method requires a generality ordering, which is monotonic with respect to pattern support. The generality ordering adopted by SPADA is based on the notion of θ-subsumption [33].

Definition 5.5. P_1 is more general than P_2 under θ-subsumption ($P_1 \succeq_\theta P_2$) if and only if P_1 θ-subsumes P_2, i.e., a substitution θ exists, such that $P_1\theta \subseteq P_2$.

For instance, with reference TO the atomsets P_1, P_2 and P_3 reported above, we observe that P_1 θ-subsumes P_2 ($P_1 \succeq_\theta P_2$) and P_2 θ-subsumes P_3 ($P_2 \succeq_\theta P_3$) with substitutions $\theta_1 = \theta_2 = \emptyset$.

The relation $\succeq_\theta$ is a quasi-ordering (or preorder), since it is reflexive and transitive but not antisymmetric. Moreover, it is monotonic with respect to support [24], as stated in the following proposition.

Proposition 5.1. *Let P_1 and P_2 be two atomsets at the same level l, defined as in Definition 5.1. If $P_1 \succeq_\theta P_2$, then $sup(P_1, D) \geq sup(P_2, D)$.*

It is noteworthy that if $P_1 \succeq_\theta P_2$ and P_1 is not frequent ($sup(P_1, D) < \sigma_{min}[l]$), then also P_2 is not frequent ($sup(P_2, D) < \sigma_{min}[l]$). This monotonicity property of $\succeq_\theta$ with respect to the support allows for pruning the search space without losing frequent atomsets.

In the inter-level search, atomsets discovered at level l are refined by descending the generalization hierarchies up to finding task-relevant objects mapped at level $l + 1$. These are the only candidate atomsets considered for evaluation, since other candidates would not meet the necessary condition for atomsets to be frequent at level $l + 1$ when $\sigma_{min}[l + 1] \leq \sigma_{min}[l]$ (see Definition 5.4). This way, the search space at level $l + 1$ is heavily pruned. Moreover, information on the units of analysis covered by atomsets at level l can be used to make more efficient the evaluation of the support of atomsets at level $l + 1$. Indeed, if a unit of analysis $D[s]$ is not covered by a pattern P at granularity level l, then it will not be covered by any descendant of P at level $l + 1$.

Once frequent atomsets have been generated at level l, it is possible to generate *strong* spatial association rules, i.e., rules whose confidence is higher than a threshold $\kappa_{min}[l]$. In particular, each frequent atomset P at level l is partitioned into two atomsets A and C such that $P = A \wedge C$ and the confidence of the association rule $A \Rightarrow C$ is computed. Different partitions of P generate different association rules. Those association rules with confidence lower than $\kappa_{min}[l]$ are filtered out.

We conclude by observing that in real-world applications a large number of frequent atomsets and strong association rules can be generated, most of which are uninteresting. This is also true for the module discovery problem (e.g., constituent motifs with a large inter-motif distance). To prevent this, some pattern constraints can be expressed in a declarative form and then used to filter out uninteresting atomsets or spatial association rules [4].

5.4 Implementation

The development of a module discovery tool effectively usable by biologists demands for the solution of several problems, both methodological and architectural. Methodological problems involve data pre-processing, namely discretization of numerical data, and the automated selection of some critical parameters such as minimum support. Architectural problems concern the interface of the tool with the external world, either to acquire data and parameters or to communicate results. In this section, solutions to these problems are briefly reported.

5.4.1 Choosing the Minimum Support Threshold

Setting up the minimum support threshold σ_{min} is not a trivial problem for a biologist when assuming no a priori knowledge about structural and functional features

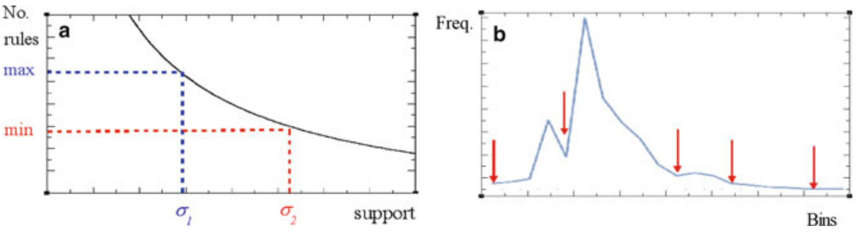


Fig. 5.3 (**a**) Functional dependence of the number of spatial association rules from the minimum support threshold. (**b**) Histogram and minimum points

Algorithm 1 Automated setting of σ_{min}

```
find_minsup($i$,[$\sigma_1, \sigma_2$],[$min, max$])
if $i \geq MAX\_ITERS$ then
    return $(\sigma_1 + \sigma_2)/2$
end if
$no\_Rules \leftarrow SPADA((\sigma_1 + \sigma_2)/2)$
if $no\_Rules \geq max$ then
    $\sigma_{min} \leftarrow find\_minsup(i + 1, [\sigma_1, (\sigma_1 + \sigma_2)/2],[min, max])$
else if $no\_Rules \leq min$ then
    $\sigma_{min} \leftarrow find\_minsup(i + 1, [(\sigma_1 + \sigma_2)/2, \sigma_2],[min, max])$
else
    $\sigma_{min} \leftarrow (\sigma_1 + \sigma_2/2)$
end if
return $\sigma_{min}$
```

of potential modules. For this reason, we follow the approach suggested in [23]: users are asked to choose an interval $[min, max]$ for the number of association rules they want to examine, and a value for σ_{min} is then automatically derived. Indeed, the number of association rules generated by SPADA depends on σ_{min} according to some function ϕ, which is monotonically decreasing (Fig. 5.3a). Therefore, the selection of an interval $[min, max]$ for the number of association rules corresponds to the selection of an interval $[\sigma_1, \sigma_2]$ for the support, which includes the optimal value σ_{min}.

Contrary to [23], where a linear search of the optimal value is proposed, we apply a dichotomic search for efficiency reasons. The formulation of the algorithm is recursive (see Algorithm 1). Initially, the procedure *find_minsup* is invoked on the support interval [0, 1] and SPADA is run with $\sigma_{min} = 0.5$. If necessary, *find_minsup* is recursively invoked on either $[0, 0.5]$ or $[0.5, 1]$. Since the convergence of the algorithm cannot be proven, we stop the search when the number of recursive invocations exceeds a maximum iteration threshold *MAX_ITERS*. A reasonable setting is $MAX\_ITERS = 5$, since after five iterations, the width of the interval $[\sigma_1, \sigma_2]$ is relatively small $(\frac{1}{2^5})$.

5.4.2 Discretizing Numerical Values

SPADA cannot properly deal with numerical values of inter-motif distances. Therefore, it is necessary to transform them into categorical values through some discretization technique. The equal frequency (EF) discretization algorithm partitions the initial range of values into a fixed number of intervals (or *bins*), such that they have different width but approximately the same number of values. This partitioning may significantly affect the subsequent rule mining step, but unfortunately, choosing a suitable number of bins is by no means an easy task for a biologist. For this reason, we investigated a new algorithm which, similarly to EF, partitions the initial range according to data distribution, but, differently from EF, it needs no input parameter.

This algorithm, called DUDA (**D**ensity-based **U**nsupervised **D**iscretization **A**lgorithm), is mainly inspired by clustering algorithms based on kernel density estimation [17], which groups together data that follow the same (typically normal) distribution (see Algorithm 2). Histograms are used to model the distribution of numerical data (inter-motif distances). The width w of each bin is computed by resorting to Scott's formula [40] $w = \frac{3.5 \times s}{\sqrt[3]{n}}$, where n is the number of values to discretize and s is the standard deviation of the values.

In this work, we look for bins so that the values in each bin are normally distributed. Partitions are identified by finding relative minimums in the histogram of frequency distribution (Fig. 5.3b), which are candidate split points for the partitioning.

Once the initial partitioning is defined, the algorithm works iteratively: at each iteration, it tries to merge two consecutive bins. Merging is performed when the distribution of values in the partition obtained after merging fits a normal distribution better than the two original bins. The decision of merging is based on the Kolmogorov–Smirnov normality test, which typically works by verifying the null

Algorithm 2 DUDA: Density-based Discretization Algorithm

```
DUDA($P$,$F$)
if $number\_of\_partitions(P) > 1$ then
  $bestL \leftarrow 0$
  for $(a, b) \in get\_consecutive\_partitions(P)$ do
    if $L_{a,b} < bestL$ then
      $(best\_a, best\_b) \leftarrow (a, b)$
      $bestL \leftarrow L_{a,b}$
    end if
  end for
  if $bestL < 0$ then
    return DUDA($merge(best\_a, best\_b, P), mergeF(best\_a, best\_b, F)$)
  end if
end if
return $P$
```

hypothesis "H_0: data are normally distributed," given a confidence level. In our case, we find the minimum confidence level α for which H_0 is not rejected, and we use it to identify the best merging according to the following formula:

$$L_{a,b} = \alpha_{a,b} \cdot (F_a + F_b) - (\alpha_a \cdot F_a + \alpha_b \cdot F_b), \tag{5.7}$$

where:

- F_a (F_b) is the relative frequency of values in the partition a (b)
- α_a, α_b and $\alpha_{a,b}$ are the confidence values of the Kolmogorov–Smirnov test on a, b and on the partition obtained after merging a and b, respectively;
- $L_{a,b}$ is the *loss* obtained after merging a and b.

Obviously, the smaller $L_{a,b}$, the better. The iteration stops when all possible $L_{a,b}$ are positive (no improvement is obtained) or no further merging is possible. The algorithm is recursive: it takes as input the list of partitions and the list of frequencies and returns a new list of partitions. The functions *merge* and *mergeF* take as input a list of r elements and return a list of $r-1$ elements, where two consecutive elements are appropriately merged.

5.4.3 Data Acquisition and Result Processing

SPADA has been integrated in a system which takes the set T of sequences from a text file. This file is processed in order to mine frequent motif sets as presented in Sect. 5.2.1. The output of this first step is an XML file which is stored in an XML repository. The corresponding document type definition (DTD) is shown in Fig. 5.4. For each frequent motif set S ($|t_S| \geq \tau_{min}$), the XML file describes the support set t_S together with some simple statistics (e.g., the ratio $|t_S|/|T|$). The module that implements the discretization algorithm DUDA (see Sect. 5.4.2) operates on data stored in the XML repository.

A wrapper of SPADA loads XML data in the extensional part D_E of the deductive database D used by SPADA itself, while rules of the intensional part D_I can be

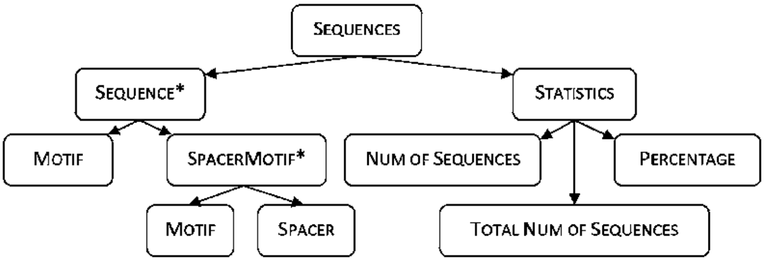


Fig. 5.4 Hierarchical structure arrangement of elements of the XML document type definition

edited by the user through a graphical user interface. This wrapper is also in charge of automatically setting the σ_{min} parameter as per Algorithm 1 in Sect. 5.4.1.

By merging consecutive bins through the rules in D_I, many spatial association rules are discovered, which differ only in some intervals of inter-motif distances. An unmanageably large number of association rules makes interpretation of results cumbersome for the biologist. For this reason, association rules are filtered before being shown to the user. Three filtering criteria are considered. The first criterion selects the association rules with the smallest bins among rules with the same motifs, the same confidence and supported by the same sequences. The second criterion selects the association rules with the greatest support among those with the same motifs and confidence, whose bins are included in the bins of the selected rules and whose list of supporting sequences is included in the list of supporting sequences of the selected rules. The last criterion selects the association rules with highest confidence among those with the same motifs, whose bins are included in the bins of the selected rules and whose list of supporting sequences is included in the list of supporting sequences of the selected rules.

5.5 Case Study

To show the potential of the integrated system, a pilot study is conducted on translation regulatory motifs located in the nucleotide sequences of untranslated regions (UTRs) of nuclear transcripts (mRNAs) targeting mitochondria. These motifs are essential for mRNA subcellular localization, stability and translation efficiency [50]. Evidence from recent studies supports the idea that the nature and distribution of these translation regulatory motifs may play an important role in the differential expression of mRNAs [11].

Datasets are generated as a view on three public biological databases, namely MitoRes,[6] UTRef and UTRsite.[7] The view integrates data on UTR sequences and their contained motifs, together with information on the motifs width and their starting and ending position along the UTR sequences in the UTRminer [48] database. We base our analysis on a set T of 728 $3'$UTR sequences relative to the human species. Twelve motifs are initially considered (set M). By setting $\tau_{min} = 4$, several frequent motif sets (set $\mathcal{S}$) are extracted in the first phase. We focus our attention on the motif set $S \in \mathcal{S}$ with the largest support set (111 $3'$UTR sequences). It contains three motifs, which are denoted as x, y and z. The hierarchy defined on motifs has three levels (Fig. 5.2), but we consider only the middle level, since the top level conveys little information on the constituent motifs of a module, while the bottom level is too specific to find interesting rules.

[6] http://www2.ba.itb.cnr.it/MitoRes/

[7] http://utrdb.ba.itb.cnr.it/

To discretize inter-motif distances, both EF and DUDA discretization are tested with two settings of the threshold σ_{min} (40% and 50%) and one of κ_{min} (80%). The number of intervals set for EF is 12. Since we have no prior knowledge on the suitability of this choice, we intentionally define some distance predicates whose semantics correspond to a merging operation of consecutive intervals (rules (5.3) reported in Sect. 5.2.2 exemplify intensionally defined distance predicates for intervals merging). This way, the comparison between EF and the discretization method proposed in this chapter is fair and does not depend on our initial decision of partitioning the distances in 12 intervals.

Experimentally, we observe that the running time varies significantly between the two solutions (Table 5.1). Indeed, the use of intensionally defined predicates to merge intervals slows down the discovery process and has the undesirable effect of returning a large number of similar rules which have to be finally filtered out.

We also test the procedure for the automated selection of the σ_{min} threshold. The interval chosen for the number of spatial association rules is $[min = 50, max = 100]$, while $MAX\_ITERS = 6$. After five steps, the system converges to $\sigma_{min} = 0.5313$ and returns 85 spatial association rules (Table 5.2).

An example of spatial association rule discovered by SPADA is the following:

$$\begin{aligned} &sequence(T), part\_of(T, M_1), is\_a(M_1, x), distance(M_1, M_2, [-99..-18]),\\ &\quad is\_a(M_2, y), distance(M_2, M_3, [-99..3.5]), M_1 \neq M_2, M_1 \neq M_3, M_2 \neq M_3\\ &\quad \Rightarrow is\_a(M_3, z) \end{aligned} \tag{5.8}$$

This rule can be interpreted as follows: if a motif of type x is followed by a motif of type y, their inter-motif distance falls in the interval $[-99..-18]$, and the motif of type y is followed by another motif at an inter-motif distance which falls in the interval $[-99..3.5]$, then that motif is of type z. The support of this rule is 63.96%, while the confidence is 100%. The high support reveals a statistically overrepresented module, which may be indicative of an associated biological function. This module can also be represented by the following chain:

Table 5.1 Results for the two discretization algorithms

	σ_{min}	Running time	No. of unfiltered rules	No. of filtered rules
Equal frequency	40%	>36 h	1,817	84
	50%	>4 h	220	36
DUDA	40%	4 s	16	16
	50%	1 s	12	12

Table 5.2 Choosing the minimum support threshold

Iteration no.	1	2	3	4	5
No. rules	185	9	25	40	85
σ_r	0.5000	0.7500	0.6250	0.5625	0.5313

$$\langle x, [-99..-18], y, [-99..3.5], z\rangle,$$

which is similar to that reported as (5.1) in Sect. 5.2.2, with the difference that here intervals of inter-motif distances are reported. The high confidence means that when the conditions expressed in the antecedent hold, the type z of the third motif in the chain can be predicted with certainty. Therefore, the spatial association rule conveys additional inferential information with respect to the frequent pattern.

5.6 Conclusions

In this chapter, we describe a new approach to module discovery by mining spatial association rules from a set of biological sequences where type and position of regulatory single motifs are annotated. The method is based on an ILP formulation which facilitates the representation of the biological sequences by means of sets of Datalog ground atoms, the specification of background knowledge by means of Datalog rules and the formulation of pattern constraints by means of a declarative formalism. Although results of the method are easy to read for a data-mining expert, they are not intelligible for a biologist because of the use of first-order logic to represent spatial patterns. For this reason, the spatial association rule mining method has been implemented in a tool which effectively support biologists in module discovery tasks by graphically rendering mined association rules. The tool also supports biologists in other critical decisions, such as selecting the minimum support threshold. To face the hard discretion problem, which typically affects discrete approaches like that described in this chapter, we have also implemented a new discretization method, which is inspired by kernel density estimation-based clustering and needs no input parameters.

The tool has been applied to a pilot study on translation regulatory motifs located in the untranslated regions of messenger RNAs targeting mitochondria. The application shows the potential of the approach and methods proposed in this chapter.

Acknowledgements This work is supported in partial fulfillment of the research objectives of two Italian projects funded by the MIUR (Italian Ministry for Education, University and Research): “LIBi: Laboratorio Internazionale di Bioinformatica” (FIRB project) and “MBLab: Laboratorio di Bioinformatica per la Biodiversità Molecolare” (FAR project). The authors gratefully acknowledge Lynn Rudd for reading the initial version of this chapter.

References

1. Aerts, S., Loo, P.V., Thijs, G., Moreau, Y., Moor, B.D.: Computational detection of cis-regulatory modules. In: Proc. of the European Conf. on Computational Biology (ECCB), pp. 5–14 (2003)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. of the 21st Int. Conf. on Very Large Data Bases, pp. 487–499 (1994)

3. Agrawal, R., Srikant, R.: Mining sequential patterns. In: P.S. Yu, A.L.P. Chen (eds.) Proc. of the 11th Int. Conf. on Data Engineering (ICDE), pp. 3–14. IEEE Computer Society (1995)
4. Appice, A., Berardi, M., Ceci, M., Malerba, D.: Mining and filtering multi-level spatial association rules with ares. In: M.S. Hacid, N.V. Murray, Z.W. Ras, S. Tsumoto (eds.) Foundations of Intelligent Systems, 15th Int. Symposium, ISMIS 2005, *LNCS*, vol. 3488, pp. 342–353. Springer (2005)
5. Bailey, T.L., Elkan, C.: Fitting a mixture model by expectation maximization to discover motifs in biopolymer. In: R.B. Altman, D.L. Brutlag, P.D. Karp, R.H. Lathrop, D.B. Searls (eds.) Proc. of the 2nd Int. Conf. on Intelligent Systems for Molecular Biology (ISMB), pp. 28–36. AAAI (1994)
6. Bi, C.: Seam: a stochastic EM-type algorithm for motif-finding in biopolymer sequences. Journal of Bioinformatics and Computational Biology **5**(1), 47–77 (2007)
7. Blockeel, H., Sebag, M.: Scalability and efficiency in multi-relational data mining. SIGKDD Explorations **5**(1), 17–30 (2003)
8. Buhler, J., Tompa, M.: Finding motifs using random projections. Journal of Computational Biology **9**(2), 225–242 (2002)
9. Ceri, S., Gottlob, G., Tanca, L.: Logic programming and databases. Springer, New York (1990)
10. Dehaspe, L., De Raedt, L.: Mining association rules in multiple relations. In: the 7th Int. Workshop on Inductive Logic Programming, ILP 1997, vol. 1297, pp. 125–132. Springer (1997)
11. Didiano, D., Hobert, O.: Molecular architecture of a miRNA-regulated 3'UTR. RNA (New York) **14**(7), 1297–1317 (2008)
12. Erman, B., Cortes, M., Nikolajczyk, B., Speck, N., Sen, R.: Ets-core binding factor: a common composite motif in antigen receptor gene enhancers. Molecular and Cellular Biology **18**(3), 1322–1330 (1998)
13. Frith, M.C., Hansen, U., Weng, Z.: Detection of cis-element clusters in higher eukaryotic DNA. Bioinformatics **17**(10), 878–889 (2001)
14. Gupta, M., Liu, J.S.: De novo cis-regulatory module elicitation for eukaryotic genomes. Proc. National Acadademy of Science **102**(20), 7079–7084 (2005)
15. Heinemeyer, T., Wingender, E., Reuter, I., Hermjakob, H., Kel, A.E., Kel-Margoulis, O.V., Ignatieva, E.V., Ananko, E.A., Podkolodnaya, O.A., Kolpakov, F.A., Podkolodny, N.L., Kolchanov, N.A.: Databases on transcriptional regulation: TRANSFAC, TRRD and COMPEL. Nucleic Acids Research **26**(1), 362–367 (1998)
16. Helft, N.: Inductive generalization: a logical framework. In: I. Bratko, N. Lavrač (eds.) Progress in Machine Learning, pp. 149–157. Sigma Press, Wilmslow (1987)
17. Hinneburg, A., Keim, D.A.: A general approach to clustering in large databases with noise. Knowledge and Information Systems **5**(4), 387–415 (2003)
18. Ivan, A., Halfon, M., Sinha, S.: Computational discovery of cis-regulatory modules in drosophila without prior knowledge of motifs. Genome Biology **9**(1), R22 (2008)
19. Jackups, R., Liang, J.: Combinatorial analysis for sequence and spatial motif discovery in short sequence fragments. IEEE/ACM Trans. Comput. Biology Bioinform. **7**(3), 524–536 (2010)
20. Johansson, Ö., Alkema, W., Wasserman, W.W., Lagergren, J.: Identification of functional clusters of transcription factor binding motifs in genome sequences: the mscan algorithm. Bioinformatics **19 (suppl 1)**, i169–i176 (2003)
21. Klepper, K., Sandve, G.K., Abul, O., Johansen, J., Drabløs, F.: Assessment of composite motif discovery methods. BMC Bioinformatics **9**, 123 (2008)
22. Li, M., Ma, B., Wang, L.: Finding similar regions in many sequences. Journal of Computer and System Sciences **65**(1), 73–96 (2002)
23. Lin, W., Alvarez, S.A., Ruiz, C.: Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery **6**(1), 83–105 (2002)
24. Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning **55**(2), 175–210 (2004)
25. Liu, X., Brutlag, D.L., Liu, J.S.: Bioprospector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. In: Pacific Symposium on Biocomputing, pp. 127–138 (2001)

26. MacIsaac, K.D., Fraenkel, E.: Practical strategies for discovering regulatory DNA sequence motifs. PLoS Compututational Biology **2**(4), e36 (2006)
27. Malerba, D., Lisi, F.A.: An ILP method for spatial association rule mining. In: In Working notes of the First Workshop on Multi-Relational Data Mining, pp. 18–29 (2001)
28. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery **1**(3), 241–258 (1997)
29. Mitchell, T.: Machine Learning. McGraw-Hill, NY (1997)
30. Muggleton, S., Srinivasan, A., King, R.D., Sternberg, M.J.E.: Biochemical knowledge discovery using inductive logic programming. In: S. Arikawa, H. Motoda (eds.) Discovery Science, *LNCS*, vol. 1532, pp. 326–341. Springer, Berlin (1998)
31. Nienhuys-Cheng, S.H., De Wolf, R.: Foundations of Inductive Logic Programming, *LNAI*, vol. 1228. Springer, Berlin (1997)
32. Perdikuri, K., Tsakalidis, A.K.: Motif extraction from biological sequences: Trends and contributions to other scientific fields. In: Proc. of the 3rd Int. Conf on Information Technology and Applications (ICITA), vol. 1, pp. 453–458. IEEE Computer Society (2005)
33. Plotkin, G.D.: A note on inductive generalization. Machine Intelligence **5**, 153–163 (1970)
34. Remnyi, A., Schler, H.R., Wilmanns, M.: Combinatorial control of gene expression. Nature Structural & Molecular Biology **11**(9), 812–815 (2004)
35. Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm [published erratum appears in bioinformatics 1998;14(2): 229]. Bioinformatics **14**(1), 55–67 (1998)
36. Robin, S., Rodolphe, F., Schbath, S.: DNA, Words and Models: Statistics of Exceptional Words. Cambridge University Press, London (2005)
37. Sandelin, A., Alkema, W., Engström, P.G., Wasserman, W.W., Lenhard, B.: JASPAR: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Research **32**(Database-Issue), 91–94 (2004)
38. Sandve, G.K., Drabløs, F.: Generalized composite motif discovery. In: R. Khosla, R.J. Howlett, L.C. Jain (eds.) Knowledge-Based Intelligent Information and Engineering Systems, 9th Int. Conf., KES 2005, vol. 3, *LNCS*, vol. 3683, pp. 763–769. Springer (2005)
39. Sandve, G.K., Abul, O., Drabløs, F.: Compo: composite motif discovery using discrete models. BMC Bioinformatics **9** (2008)
40. Scott, D.: On optimal and data-based histograms. Biometrika **66**, 605–610 (1979)
41. Segal, E., Sharan, R.: A discriminative model for identifying spatial cis-regulatory modules. Journal of Computational Biology **12**(6), 822–834 (2005)
42. Sharan, R., Ovcharenko, I., Ben-Hur, A., Karp, R.M.: CREME: a framework for identifying cis-regulatory modules in human-mouse conserved segments. Bioinformatics **19 (suppl 1)**(18), S283–S291 (2003)
43. Sinha, S., Tompa, M.: A statistical method for finding transcription factor binding sites. In: P.E. Bourne, M. Gribskov, R.B. Altman, N. Jensen, D.A. Hope, T. Lengauer, J.C. Mitchell, E.D. Scheeff, C. Smith, S. Strande, H. Weissig (eds.) ISMB, pp. 344–354. AAAI (2000)
44. Srinivasan, A., King, R.D., Muggleton, S., Sternberg, M.J.E.: Carcinogenesis predictions using ILP. In: N. Lavrac, S. Dzeroski (eds.) Inductive Logic Programming, 7th International Workshop, ILP-97, *LNCS*, vol. 1297, pp. 273–287. Springer (1997)
45. Srinivasan, A., King, R.D., Muggleton, S., Sternberg, M.J.E.: The predictive toxicology evaluation challenge. In: Proc. of the 15th Int. Joint Conf. on Artificial Intelligence (IJCAI), pp. 4–9 (1997)
46. Stormo, G.D.: DNA binding sites: representation and discovery. Bioinformatics **16**(1), 16–23 (2000)
47. Takusagawa, K.T., Gifford, D.K.: Negative information for motif discovery. In: R.B. Altman, A.K. Dunker, L. Hunter, T.A. Jung, T.E. Klein (eds.) Pacific Symposium on Biocomputing, pp. 360–371. World Scientific, Singapore (2004)
48. Turi, A., Loglisci, C., Salvemini, E., Grillo, G., Malerba, D., D'Elia, D.: Computational annotation of UTR cis-regulatory modules through frequent pattern mining. BMC Bioinformatics **10 (suppl 6)**, S25 (2009)

49. Valiant, L.G.: A theory of the learnable. Communications of the ACM **27**(11), 1134–1142 (1984)
50. Wilkie, G., Dickson, K., Gray, N.: Regulation of mRNA translation by 5'- and 3'-UTR-binding factors. Trends in Biochemical Sciences **28**(4), 182–188 (2003)
51. Xing, E.P., Wu, W., Jordan, M.I., Karp, R.M.: Logos: a modular bayesian model for de novo motif detection. Journal of Bioinformatics and Computational Biology **2**(1), 127–154 (2004)
52. Zhou, Q., Wong, W.H.: CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling. Proceedings of the National Academy of Sciences of the United States of America **101**(33), 12114–12119 (2004)

Chapter 6
Modeling Biochemical Pathways

Ettore Mosca and Luciano Milanesi

Abstract Sequence analysis methods predict macromolecule properties and intermolecular interactions. These data can be used to reconstruct molecular networks, which are complex systems that regulate cell functions. Systems biology uses mathematical modeling and computer-based numerical simulations in order to understand emergent properties of these systems. This chapter describes the approaches to define kinetic models to simulate biochemical pathways dynamics. It deals with three main steps: the definition of the system's structure, the mathematical formulation to reproduce the time evolution and the parameter estimation to find the set of parameter values such that the model behavior fits the experimental data.

6.1 Introduction

Sequence analysis provides a series of information related both to macromolecules and to their intermolecular interactions. In fact, the analysis of DNA, RNA and protein sequences allows us to infer some of the sequence's properties: for instance, it is possible to identify coding and non-coding regions in a genome and to predict the secondary and the tridimensional structure assumed by RNAs and proteins. Sequence analysis also plays an important role when inferring physical interactions between these molecular components. Computational methods exist for the identification of RNA- and DNA-binding proteins: for instance, this is the case of the transcription factors–DNA interactions, which provide knowledge about the genes regulated by specific transcription factors. Another interesting domain of application is the binding between non-coding RNAs (RNAs that are not translated into proteins) and other types of RNAs: for instance, microRNAs (also referred to as miRNAs or μRNAs) bind to mRNAs (messenger RNAs) preventing the translation.

L. Milanesi (✉)
Institute for Biomedical Technologies, CNR, Milan, Italy
e-mail: luciano.milanesi@itb.cnr.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_6,

The integration of data provided by sequence analysis approaches yields important knowledge for the reconstruction of the “molecular circuits,” which exploit metabolic (related to mass and energy flows), signal transduction (concerning the transmission of signals) and gene regulatory processes (describing the regulation of gene expression). The understanding of the structural and dynamical properties of molecular networks is mainly important since biological functions arise from the activity of these circuits. In order to reach this goal, it is necessary to define mathematical models that permit numerical simulations; in fact, analytical solutions are available only in the few cases in which the modeled pathway is composed of a small number of molecular species.

In this chapter, we will discuss the definition of mathematical models for the analysis of molecular pathways. Models can be defined starting from data provided by sequence analysis methods and represent a relevant application of these data to shed light on the functioning of living organisms.

Section 6.2 introduces systems biology and underlies the importance of the quantitative study of biological systems; Section 6.3 describes the steps involved in building a mathematical model; Section 6.4 illustrates the representation of molecular circuits structure and its analysis; Section 6.5 describes the modeling approaches that can be used to study the system time evolution; Section 6.6 copes with the problem of the estimation of model parameters; Section 6.7 summarizes the message of the chapter.

6.2 Complexity of Living Organisms and Systems Biology

Living organisms are intrinsically complex. This concept can be easily visualized considering the molecular scale, in which biological processes are controlled by the activity of intricate and heterogeneous regulatory networks established by the interactions among DNA, RNAs, proteins, metabolites and other organic and inorganic elements. These networks exhibit complex dynamics and, in general, regulate biological processes determining a response that may vary across many orders of magnitudes [24]. Moreover, molecular circuits are sensitive to environmental changes and are spatially regulated by means of the intracellular structures. The complexity increases considering that cells organize in tissues, tissues in organs and organs in the whole organism, and all these scales control biological functions. Biological systems have been tuned by the evolution process to be robust, but also to be evolvable (for instance, DNA replication is a process with an astonishing efficiency, but some errors occur and are crucial for life evolvability). During evolution this phenomenon generated complex and very improbable structures, like the so-called bowtie architecture [8] of the molecular processes.

Complex systems show *emergent properties*. This properties can appear when a number of elements operate in an environment, resulting in more complex behaviors as a collective. Considering living systems, swarms show complex behaviors manifesting structures, patterns and properties during the process of self-organization [7], while, at the molecular level, the presence of feedback loops in a biochemical system can determine oscillations of some of the species concentrations [25]. The concept of emergent properties refers to two situations:

- *Weak emergence*: The phenomenon wherein complex, interesting high-level function is produced as a result of combining simple low-level mechanisms in simple ways; systems showing this kind of emergence are, for example, the game of life and biochemical systems [5]
- *Strong emergence*: A phenomenon that arises from the low-level domain, but truths concerning that phenomenon are not deducible even in principle from truths in the low-level domain [5].

The reductionist approach aims to understand the reality by the study of its constituents. Reductionism led to remarkable results related to the knowledge of molecular components of biological systems, such as genes, RNAs, proteins, metabolites and biochemical reactions. However, properties of many systems resist to a reductionist explanation [23], and the failure of the reductionist approach is mainly related to the complexity of biological systems. Therefore, while the study of the "building-blocks" is still important, a system level approach, aimed to the understanding of system structures, system dynamics, the control method and the design method, is fundamental for a deeper understanding of biological processes [14]. *Systems biology* is the multidisciplinary field which pursues this goal using knowledge and approaches from a series of disciplines such as biology, chemistry, physics, computer science, mathematics and engineering. Historically, it is possible to identify two roots which have led to the approach referred to as systems biology [27]: on the one hand, the evolution of molecular biology; on the other hand, the formal analysis of molecule systems.

6.3 Steps Involved in the Definition of a Kinetic Model

Systems biology studies systems described by means of the interactions of organic and inorganic components that reside within the cell and its environment. A model is usually developed by following a process structured in three sequential steps:

1. The definition of the wiring diagram and the structure of the system;
2. The mathematical formulation for the system's dynamics;
3. The identification of a set of proper values for the model parameters, in order to obtain a behavior comparable to experimental data or to some known dynamics.

It is important to consider that there is no unique correspondence between a wiring diagram and the mathematical formulation. The same biological mechanism can be represented by different forms of equations. Since these choices are somewhat arbitrary, there is a hierarchy of assumptions associated with the model definition, from the assembly of the wiring diagram to the assignment of specific values to the parameters appearing in the mathematical formulation. Once the model has been developed, parameter values are adjusted to try to fit the experimental data. If this step is too complicated, one may reconsider the assumptions made during the previous steps: for instance, some molecular entities or biochemical processes have not been considered and their role may be relevant for the system dynamics.

6.4 The Wiring Diagram and the System Structure

The first step during model development concerns the definition of the *wiring diagram*, in which a set of boxes (components) are interconnected by arrows (biochemical processes). This diagram captures the functioning of the modeled biological process in terms of the molecular entities controlling the process and the interactions established. These interactions represent biochemical processes and include transient intermolecular interactions (e.g., the association of two proteins to form a protein complex), biochemical reactions and the movements between different spatial locations (e.g., movement from the cytosol to the nucleus). The *systems biology graphical notation* (SBGN) [20] is a visual language developed by the scientific community in order to standardize the development of wiring diagrams. It specifies the connectivity of the graphs and the types of the nodes and edges. In particular, SBGN defines three complementary specifications to create wiring diagrams:

- The *process diagram*, which considers all the molecular processes and interactions taking place between biochemical entities and their results;
- The *entity relationship diagram*, which puts the emphasis on the influences that entities have upon each other's transformations rather than the transformations themselves;
- The *activity flow diagram*, which allows modulatory arcs to directly link different activities, rather than entities and processes or relationships.

The process diagram provides the more detailed and unambiguous representation of the pathway, supporting also the temporal sequentiality between events; unfortunately, the process diagram is sensitive to combinatorial explosion of states and processes. The entity relationship diagram reduces such explosion by representing physical entities only once; however, this notation does not support sequentiality between events, and some biochemical processes (e.g., creation and destruction of a molecular entity) are not easily represented. The activity flow diagram is the more essential representation; it provides a conceptual description of influences, being the more compact, and ambiguous, representation.

An example of SBGN *process diagram* is reported in Fig. 6.1. The molecular circuit considers the set of species $S = \{\emptyset, s_1, s_2, \ldots, s_9\}$. s_1 represents a transcription factor that promotes the expression of the proteins s_2 and s_3. These two proteins are enzymes that control the biochemical reactions chain from s_4 to s_8. The circuit contains a feedback control, due to the binding of s_8 to s_1, leading to a decrease in the s_2 concentration. The entity $\emptyset$ denotes a source or a sink and constitutes the system

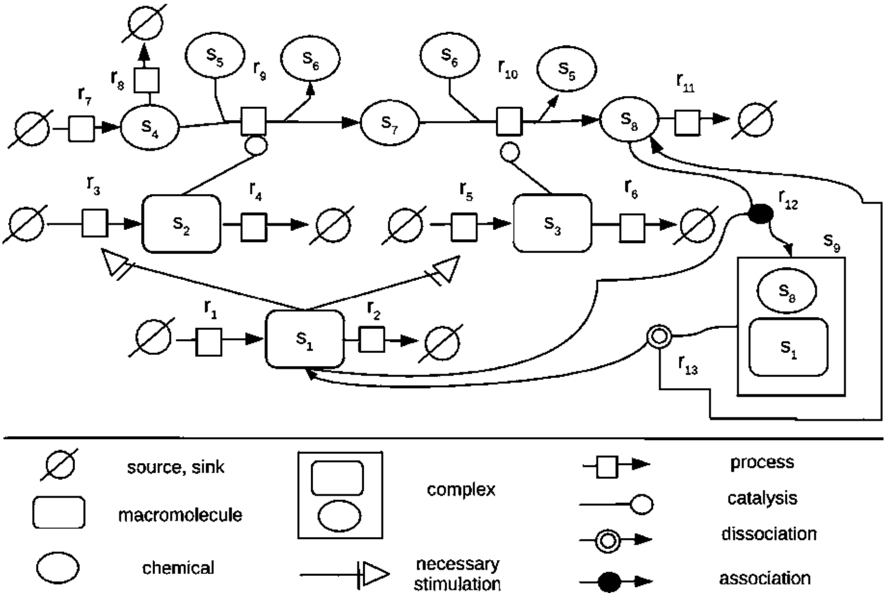


Fig. 6.1 Process diagram of an hypothetical molecular circuit controlled by a transcription factor s_1, which is necessary to activate the two enzymes s_2 and s_3, that control the biochemical reactions chain transforming s_4 into s_8; $r_1, r_2, \ldots, r_{13}$ indicates the biochemical processes

boundary conditions. Note that the regulatory interactions which s_1 establishes with s_2 and s_3 can be predicted with sequence analysis methods for the identification of transcription factor-binding site on the genome.

Once the wiring diagram has been drawn, it is possible to formalize the biochemical network. In general, the molecular species $S = \{s_1, \ldots, s_n\}$ and biochemical processes $R = \{r_1, \ldots, r_m\}$ can be organized in a weighted directed graph $D = (S \cup R, E)$, where the nodes are the molecular species and reactions, while the directed arcs, $E \subseteq ((S \times R) \cup (R \times S))$, connect reactants to reactions and reactions to products. Arc weights are the *stoichiometric coefficients*, α_{ij} and β_{ij}, that are associated with each molecular species in each reaction $r_j : \alpha_{1j} s_1, \ldots, \alpha_{nj} s_n \rightarrow \beta_{1j} s_1, \ldots, \beta_{nj} s_n$. Stoichiometric coefficients establish the quantity of reactants (left side of r_j) and products (right side of r_j) which participate in the process. For example, the reactions of the biochemical system depicted in Fig. 6.1 are: $r_1 : \emptyset \rightarrow s_1$, $r_2 : s_1 \rightarrow \emptyset$, $r_3 : \emptyset \rightarrow s_2$, $r_4 : s_2 \rightarrow \emptyset$, $r_5 : \emptyset \rightarrow s_3$, $r_6 : s_3 \rightarrow \emptyset$, $r_7 : \emptyset \rightarrow s_4$, $r_8 : s_4 \rightarrow \emptyset$, $r_9 : s_4 + s_5 \rightarrow s_6 + s_7$, $r_{10} : s_6 + s_7 \rightarrow s_5 + s_8$, $r_{11} : s_8 \rightarrow \emptyset$, $r_{12} : s_1 + s_8 \rightarrow s_9$, $r_{13} : s_9 \rightarrow s_1 + s_8$; note that the species that appear in the listed reactions have stoichiometric coefficients equal to 1 and species that do not appear have $\alpha_{ij} = 0$ or $\beta_{ij} = 0$.

Stoichiometric coefficients are usually organized in the *stoichiometry matrix* $\mathbf{N} : n \times m$, whose elements $v_{ij} = \beta_{ij} - \alpha_{ij}$ indicate if the species s_i is consumed ($v_{ij} < 0$), produced ($v_{ij} > 0$) or if it does not change ($v_{ij} = 0$) due to the process r_j. The stoichiometric matrix related to the system of Fig. 6.1 is:

$$\mathbf{N} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix},$$

where the species $\emptyset$ is omitted, the rows and columns are ordered according to the indexes i, j of s_i and r_j, i.e., the first row indicates that one molecule of s_1 is produced due to r_1 and r_{13} while it is consumed due to r_2 and r_{12}.

6.4.1 Analysis of the Stoichiometric Matrix

The analysis of the stoichiometric matrix allows to study two interesting properties: the *conservation relations* and the *steady-state flux relations*. Importantly, these two properties, derived following the analysis of the system structure, predict some characteristics of the system dynamics. Depending on the values of **N**, a particular pathway can show none, one or more conservation relations and steady-state flux relations.

The *conservation relations* characterize weighted sums of molecular entities concentrations which remain constant in the system [15] and represent a combination of rows of **N** that are linearly dependent. Linear combinations of rows of **N** can be represented by $\mathbf{y}^{\mathrm{T}}\mathbf{N}$. In particular, the conservation relations must fulfil:

$$\mathbf{y}^{\mathrm{T}}\mathbf{N} = \mathbf{0}^{\mathrm{T}}$$

i.e., **y** lies in the left null-space of **N**. The conservation relations that are composed of a positive sum of metabolite concentrations are the *conserved moieties*. Considering the pathway of Fig. 6.1, it is possible to identify two conserved relations that can be arranged in the matrix **Y**:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^{\mathrm{T}} \\ \mathbf{y}_2^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

which mean, respectively:

$$x_5 + x_6 = k_1$$
$$x_5 + x_7 = k_2$$

where k_1, k_2 are constants and x_i indicates the number of molecules of the species s_i. The two conservation relations indicate that whenever one molecule of s_5 is consumed, one molecule of s_6 and one molecule of s_7 will be produced. Moreover, any linear combinations of the conservation relations are also conserved; in this case we obtain that the sum:

$$2x_5 + x_6 + x_7 = k_1 + k_2$$

does not vary during the system dynamics.

The other interesting property that we can gain from the analysis of **N** are the *flux relationships at steady state*. These relationships include the steady-state fluxes in a network that do not violate the principle of mass conservation. In order to understand this concept, let us introduce the concept of steady state. The dynamics of the molecular circuit can be represented by the following system of differential equations:

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \mathbf{Nf},$$

where $\mathbf{X}(t)$ is the vector of the species concentrations and $\mathbf{f} = (f_1, \ldots, f_m)$ is the vector of functions $f_j(\mathbf{X}(t))$, which define the fluxes of the biochemical processes R. Steady states are particular states defined as

$$\frac{\mathrm{d}\mathbf{X}(t)}{\mathrm{d}t} = \mathbf{Nf} = 0$$

i.e., where the species concentrations remain fixed during the time evolution of the system. All the possible solutions are contained in the null space of **N**; trivial solution $\mathbf{f}(\mathbf{X}(t)) = 0$ is usually not considered since it identifies the thermodynamic equilibrium and, in the case of biological systems, means death! Considering the system of Fig. 6.1, it is possible to identify six vectors that lie in the null space of **N**. These vectors can be arranged in the matrix **K**:

$$\mathbf{K} = \begin{bmatrix} 0&0&0&0&1&0 \\ 0&0&0&0&1&0 \\ 0&0&1&0&0&0 \\ 0&0&1&0&0&0 \\ 0&1&0&0&0&0 \\ 0&1&0&0&0&0 \\ 1&0&0&1&0&0 \\ 1&0&0&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&0&1 \\ 0&0&0&0&0&1 \end{bmatrix}$$

Every solution specifies the processes which have fluxes that establish relations at steady state. Observing the pathway wiring diagram in Fig. 6.1, it is easy to check that the six fluxes combinations reported in **K** do not violate the mass conservation principle. For instance, the fourth vector considers the flux among the chain of biochemical processes r_7, r_9, r_{10}, r_{11}: the species net mass variation related to the application of these reactions is null.

6.5 Modeling Approaches for the Time Evolution of Well-Stirred Systems: From the Chemical Master Equation to ODE Models

In this section, we will consider well-stirred or spatially homogeneous systems, meaning that the chemical species abundances do not vary with respect to space. This assumption seems to be hardly justified within cells, where there is a very crowded environment. However, whether it is a good approximation or not, depends on the time scale of the considered biological process. In the case in which the biological process involves a time scale that is greater than the molecules diffusion time scale, the well-stirred approximation is justified. In many cases, such as cell-cycle regulation or circadian rhythms, the well-stirred assumption is appropriate and well-stirred models have been successfully used to obtain a deeper understanding of these biological processes [1, 16].

Let us consider a well-stirred system of chemical species S interacting by means of chemical reactions R. The system's volume and temperature are constant. Each chemical reaction is characterized by two quantities. The first is the state change vector $\mathbf{v} = (v_{1,j}, \ldots, v_{n,j})^{\mathrm{T}}$, where $v_{i,j}$ is the change (in terms of molecule numbers) in the s_i population due to the reaction r_j; therefore the state change vectors form the stoichiometric matrix: $\mathbf{N} = [\mathbf{v}_1, \ldots, \mathbf{v}_m]$. The second is the *propensity function* $a(\mathbf{v})$, that is a function such that $a(\mathbf{v})\mathrm{d}t$ gives the probability that one reaction of the type r_j occurs in the next infinitesimal time interval $[t, t + \mathrm{d}\tau]$. The vector $\mathbf{x} = (x_{1,j}, \ldots, x_{n,j})^{\mathrm{T}}$, where x_i is the number of molecules of the chemical species s_i, defines the system's state. It is possible to deduce a time equation for describing the time evolution of the system's state (given a particular initial condition x_0) using the laws of probability [12]:

$$\frac{\mathrm{d}p(\mathbf{x}, t|\mathbf{x}_0, t_0)}{\mathrm{d}t} = \sum_{j=1}^{m} [a_j(\mathbf{x} - \mathbf{v}_j)p(\mathbf{x} - \mathbf{v}_j, t|\mathbf{x}_0, t_0) - a_j(\mathbf{x})p(\mathbf{x}, t|\mathbf{x}_0, t_0)],$$

where $p(\mathbf{x}, t|\mathbf{x}_0, t_0)$ is the conditional probability of the system to be at state $\mathbf{x}$ and time t given the initial state $\mathbf{x}_0$ at time t_0. This set of first-order differential equations is the so-called chemical master equation (CME). It is easy to see that the CME consists of a number of ordinary differential equations (ODEs) equal to the number of possible states. As the state values are typically unbounded, the number of ODEs is infinite. Therefore, the analytical solution is possible only in few cases and the numerical solutions are usually very computationally intensive. To overcome this limitation, Gillespie proposed the stochastic simulation algorithm (SSA) [11]. This algorithm is a Monte Carlo strategy and provides exact numerical realizations of the stochastic process defined by the CME.

6.5.1 Successive Approximations Lead to Reaction Rate Equations

However, since the SSA simulates every single reaction event, it becomes computationally hard for systems in which a large number of reactions occur during its time evolution. This situation determined the development of other approaches which are no longer exact but reduce the computational cost of a simulation of the system.

The *tau leaping technique* [4] is one of these approximated approaches. This algorithm advances the system firing more than one reaction during a pre-selected time step τ: the number of firings is obtained from a Poisson random variable $P(a_j(\mathbf{x}), \tau)$ of mean and variance equal to $a_j(\mathbf{x})\mathrm{d}t$. The value of τ is computed in order to satisfy the leaping condition, according to which the state change must be sufficiently small that no propensity function changes its value by a significant amount. Hence, the current system's state is calculated according to the following formula:

$$\mathbf{X}(t + \tau) \doteq \mathbf{x} + \sum_{j=1}^{m} \mathbf{v}_j P_j(a_j(\mathbf{X}(t)), \tau)$$

For example, considering the pathway of Fig. 6.1, the formula related to the species s_2 is:

$$X_2(t + \tau) \doteq x_2 + v_{2,3}\ P_3(a_3(\mathbf{X}(t)), \tau) + v_{2,4}\ P_4(a_4(\mathbf{X}(t)), \tau),$$

because the only null stoichiometric coefficients regarding s_2 are $v_{2,3}$ and $v_{2,4}$.

If the populations of all the reactant species are sufficiently large, each reaction is expected to fire more than one in the next τ. If this condition and the leaping condition hold, the tau leaping procedure can be approximated obtaining a stochastic differential equation, called the *chemical Langevin equation* [12]:

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \doteq \sum_{j=1}^{m} \mathbf{v}_j a_j(\mathbf{X}(t)) + \sum_{j=1}^{m} \mathbf{v}_j \sqrt{a_j(\mathbf{X}(t))}\ \Gamma_j(t),$$

where $\Gamma_j(t)$ are statistically independent "Gaussian white noise" processes. Here, the state of the system $\mathbf{X}(t) = \mathbf{x}$ is a continuous random variable and is expressed as the sum of two terms: a deterministic drift term and a fluctuating diffusion term. This equation is derived approximating the Poisson random variable (integer valued) with a normal random variable (real valued) with the same mean and variance.

As the population of the chemical species increases, the second term in the above equation has a negligible effect compared with the first one: indeed, this one scales linearly with the size of the system while the second term scales only sublinearly. Leaving out the stochastic term of the above equation, the time evolution of the system can be represented by the *reaction rate equations*, a continuous and deterministic approach:

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} \doteq \sum_{j=1}^{m} \mathbf{v}_j a_j(\mathbf{X}(t))$$

Note that considering the notation of Sect. 6.4, $\sum_{j=1}^{m} \mathbf{v}_j a_j(\mathbf{X}(t)) = \mathbf{Nf}$. According to this formalism, the functions $a_j(\mathbf{X}(t))$ are called *kinetic laws* and define the rate of the biochemical reactions; moreover, the system state is usually expressed in terms of molecules concentration. A general form of kinetic laws is

$$a_j(\mathbf{X}(t)) = k_j \prod_{i=1}^{N} X_i^{\beta_i}(t),$$

where the real valued elements k_j and β_i are, respectively, the kinetic constants and the kinetic orders. Following this definition, the rate of the process r_j is determined by the kinetic constant and the product among the molecular species concentrations. For example, considering the reaction $r_9 : s_4 + s_5 \rightarrow s_6 + s_7$, a possible kinetic law is:

$$a_9 = k_9 \cdot X_2 \cdot X_4 \cdot X_5$$

i.e., the rate of production of s_6 and s_7 is determined by a specific kinetic constant, k_9, and by the concentration of reactants s_4, s_5 and enzyme s_2.

A number of different modeling formalisms arise from this general form [26]. On the one hand, if kinetic orders assume only integer values, we obtain the *conventional kinetic models*. In turn, by introducing particular approximations in this formalism, it is possible to derive a number of specific equations such as the Michaelis–Menten equation. On the other hand, if kinetic orders can be real valued, we have the class of *power law models*. If the values are only positive we have *detailed power law models*, while if the values can be both negative and positive we have *simplified power law models*. *S-Systems* ("synergism and saturation") models can be derived from simplified power law models, by the aggregation of all the positive kinetic laws ($v_{i,j} > 0$) in a unique input flux and all the negative kinetic laws ($v_{i,j} < 0$) in a unique output flux. According to this formalism each ODE is defined as follows:

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = k_i \prod_{j=1}^{n} X_j^{\beta_{i,j}} - k_i' \prod_{j=1}^{n} X_j^{\beta_{i,j}'}$$

The map depicted in Fig. 6.2 summarizes the relationships among the mathematical descriptions presented above. By introducing successive approximations, it is possible, first, to switch from a discrete and stochastic description to a continuous and stochastic one and, second, to derive a continuous and deterministic approach.

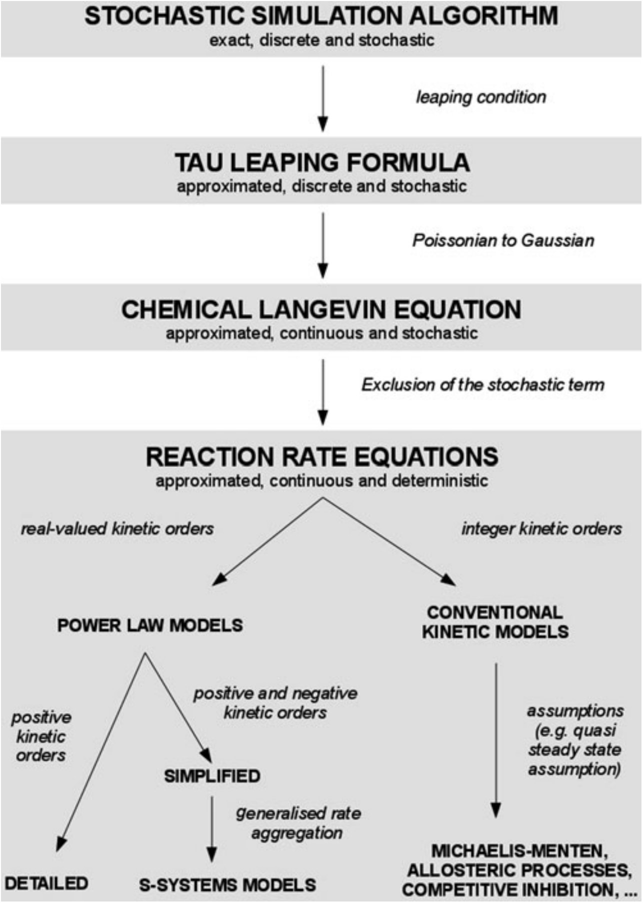

STOCHASTIC SIMULATION ALGORITHM
exact, discrete and stochastic
leaping condition
TAU LEAPING FORMULA
approximated, discrete and stochastic
Poissonian to Gaussian
CHEMICAL LANGEVIN EQUATION
approximated, continuous and stochastic
Exclusion of the stochastic term
REACTION RATE EQUATIONS
approximated, continuous and deterministic
real-valued kinetic orders
integer kinetic orders
POWER LAW MODELS
CONVENTIONAL KINETIC MODELS
positive kinetic orders
positive and negative kinetic orders
SIMPLIFIED
assumptions (e.g. quasi steady state assumption)
generalised rate aggregation
DETAILED
S-SYSTEMS MODELS
MICHAELIS-MENTEN, ALLOSTERIC PROCESSES, COMPETITIVE INHIBITION, ...

Fig. 6.2 Map of the relationships between mathematical descriptions for well-stirred systems. Tau leaping, the chemical Langevin equation and the reaction rate equation are successive approximations derived from the CME for which the SSA provides exact numerical realizations. The reaction rate equations map has been adapted from [26]

The approximations are justified by the consideration of systems with larger and larger number of molecules. Moreover, passing from the Gillespie's algorithm to ODE modeling, there is a substantial reduction in the computational cost: usually, ODEs can be easily solved in the scale of tens of seconds even for large systems (composed of tens of ODEs).

6.5.2 *One Size Does Not Fit All*

Given all these mathematical descriptions that can be used to reproduce the time evolution of a biochemical pathway, the question on the proper usage in relation to a particular biological process arises naturally. The answer is not simple and involves a series of factors that may be in contrast, such as the computational cost and the accuracy of the numerical simulations. Other factors that may influence the choice include the considered biological process, the experimental data and known computational methods. Therefore, there is no “perfect” mathematical modeling framework which can fulfil all the requirements.

One important choice concerns the use of a deterministic or a stochastic formulation of the biological system. The decision mainly depends on the number of molecules involved in the system, since the influence of the stochastic fluctuations increases as the number of molecules decreases; however, noise may influence the behavior of particular dynamical systems even with a large number of molecules [9]. If a concentration drops below the threshold of 1 nM, it may be appropriate to study the system dynamics using the stochastic approach [6]. While the stochastic and discrete approach provides a description closer to reality than the deterministic and continuous approach does, the latter relies on a broader set of developed theories and computational methods for the analysis of the system structure and dynamics.

6.6 Parameter Estimation

Once a model has been developed, a set of proper values for its parameters is requested to fit the experimental data. In fact, these values and the initial values of the model variables influence the system dynamics. The sensitivity of a model to its parameters is a property of the model itself. In general, the impact of some parameters to the model dynamics is so crucial that the nature of the dynamics produced by the model may change dramatically, determining, for instance, the appearing (or disappearing) of an oscillatory behavior.

Due to the lack of experimental measurements, experimental errors and biological variability, the value of many parameters of the models is unknown or uncertain [17]. A possible computational solution is parameter estimation, which can be informally stated as the identification of the parameter values such that the model dynamics fits the experimental data. Parameter estimation is more simple in deterministic models than stochastic models: indeed, by definition, stochastic models exhibit a series of possible different time evolutions given the same initial conditions. Deterministic models are more widespread than stochastic ones, and therefore, we focus on them.

Considering models based on a non-linear ODE system (the more common and more general case), the parameter estimation problem can be formulated as a nonlinear programming (NLP) problem with ODEs constraints:

$$J : \overbrace{[\mathbb{R}^n \ \ldots \ \mathbb{R}^n]}^{u} \times \overbrace{[\mathbb{R}^n \ \ldots \ \mathbb{R}^n]}^{u} \to \mathbb{R} \tag{6.1}$$

$$\frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t} = \mathbf{Nf}(\mathbf{p}, \mathbf{X}(t)) \tag{6.2}$$

$$\mathbf{X}(t_0) = \mathbf{x_0} \tag{6.3}$$

$$\mathbf{p}^{\mathrm{L}} \leq \mathbf{p} \leq \mathbf{p}^{\mathrm{H}}, \tag{6.4}$$

where (6.1) indicates the objective function that takes as input two $n \times u$ real valued matrices, one with the experimental data and the other with the model predictions (u is the number of time points considered), and maps them to a real value indicating their similarity; (6.1)–(6.1) are the constraints where $\mathbf{p}$ are the parameters, $\mathbf{p}^{\mathrm{L}}$ and $\mathbf{p}^{\mathrm{H}}$ are, respectively, the inferior and superior constraints over the parameter values.

Considering the example in Fig. 6.1, eight time-series data are needed, i.e., one time series for each molecular specie s_i. The number of parameters to estimate depends on how the functions $\mathbf{f}$ are written: in the case of conventional kinetic models, one parameter (the kinetic constant k_j) must be estimated for each reaction r_j, i.e., a total of 13 parameters, while kinetic orders are deduced from the reactions stoichiometry.

Due to the nonlinear and constrained nature of the system dynamics, the NLP-ODE problem is very often nonconvex. Therefore, the solutions must be searched with a global optimization (GO) method, since it is very likely that a local method would identify a solution of local nature.

GO methods can be classified in two broad sets: deterministic and stochastic. In a recent comparison among the GO methods for the parameter estimation in biochemical pathways, stochastic methods proved to be the best candidates [18]. In particular, the best solution was identified by an algorithm belonging to the class of the evolutionary computations (EC) [10]. This class of methods is very widespread and is based on the biological evolution, driven by the mechanisms of reproduction, mutation and survival of the individual with the highest fitness, which is represented by the cost function. Similar to the biological evolution, EC methods apply mutation, recombination and selection operators to a population of individuals that represent candidate solutions. Among EC methods, there are genetic algorithms (GA), evolutionary programming and evolution strategies (ES). ES are the most efficient and robust especially for GO problems with continuous variables [2, 3, 13, 21, 22]: indeed, a particular ES, the stochastic ranking evolution strategy (SRES) [21] algorithm identified the best solution in the comparison among a series of methods for GO of systems biology models [18].

SRES is a (μ,λ)-ES, where μ is the number of parents in the population of individuals, while λ indicates the number of individuals of the offspring. Every individual is an instance of the vector of parameters to be estimated ($\mathbf{p}$). The initial distribution of individuals is generated according to a uniform probability distribution over the search space, i.e., the space of all the possible solutions. The selection of the best μ individuals, which will be used to create the next generation, takes place among the λ individuals of the current generation. During the next generation creation, each μ individual is mutated, first, averaging between the values of two selected individuals and, second, updating the new values using a normally distributed one-dimensional random variable. Each individual is then used to create

λ/μ offspring on average so that a new population of λ individuals is obtained. This mechanism determines that parents survive only for one generation.

Stochastic GO approaches can provide good solutions in modest computation time, are quite simple to implement and do not require a transformation of the original problem [18]. However, large instances of the NLP-ODE problem cannot be solved in a reasonable time, even considering stochastic approaches. It is therefore important to consider strategies to face the problem of parameter estimation of systems biology ODE models on high performance computing and distributed platforms [19].

6.7 Conclusions

Living organisms are complex and must be studied considering both reductionistic and holistic approaches. At the molecular level, biological processes arise from the interactions among a number of molecular entities. Importantly, sequence analysis methods predict properties of both macromolecules and intermolecular interactions. These data are useful to construct molecular circuits that can be studied using mathematical modeling and computer-based numerical simulations.

The definition of a systems biology model begins with the assembly of the wiring diagram, which essentially capture the system structure in terms of molecular entities and interactions among entities. Graphically, the wiring diagram should be drawn according to the specifications of the SBGN. Formally, the structure of a molecular circuit can be represented by a graph and by the stoichiometric matrix. The analysis of the structure leads to two important results: the conservation relations, indicating the linear combinations of molecular entities the concentration of which do not vary during the system evolution; the flux relations at steady state, defining the biochemical processes that have the same intensity when the system is at steady state.

The time evolution of a well-stirred biochemical system is defined by the CME; unfortunately both analytical and numerical solutions are available only in a few cases. The Gillespies's algorithm provides exact numerical simulations of the stochastic process defined by the CME, but it is computationally intensive as the system size increases. The τ leaping method is an approximation that reduces the SSA computational cost. As the number of molecules included in the system increases, the τ leaping method can be approximated by the chemical Langevin equation, that is a stochastic and continuous approach. In the end, for even larger number of molecules, the reaction rate equations can be derived from the chemical Langevin equation, excluding the stochastic drift term. The reaction rate equations models (ODE models) are the most used and a lot of theories and computational methods are available for their analysis (e.g., steady-state analysis, bifurcation analysis, sensitivity analysis).

Once both, the structure and the kinetic formulation of the model are defined, a set of values for the model parameters must be found. This task is not easy due to both the number of parameters included in the model and model sensitivity to parameters value variations. Among the parameter estimation techniques, the evolutionary computations proved to be a good solution to handle this task. As the number of parameters increases, it is crucial to use high performance computing or high throughput computing to find candidate solutions to this problem.

Acknowledgements The authors are supported by the NET2DRUG, MIUR-FIRB LITBIO (RBLA0332RH), ITALBIONET (RBPR05ZK2Z), BIOPOPGEN (RBIN064YAT) and CNR-BIOINFORMATICS initiatives.

References

1. Alfieri, R., Merelli, I., Mosca, E., Milanesi, L.: A data integration approach for cell cycle analysis oriented to model simulation in systems biology. BMC Syst Biol **1**, 35 (2007). doi:10.1186/1752-0509-1-35. http://dx.doi.org/10.1186/1752-0509-1-35
2. Back, T.: Evolution strategies: an alternative evolutionary algorithm. In: Artificial evolution, Lecture Notes in Computer Science, vol. 1063, pp. 3–20. Springer, Berlin (1995)
3. Balsa-Canto, E., Alonso, A.A., Banga, J.R.: Dynamic optimization of bioprocess: deterministic and stochastic strategies. In: Proceedings of ACoFop IV (1998)
4. Cao, Y., Gillespie, D.T., Petzold, L.R.: Efficient step size selection for the tau-leaping simulation method. J Chem Phys **124**(4), 044109 (2006). doi:10.1063/1.2159468. http://dx.doi.org/10.1063/1.2159468
5. Chalmers, D.J.: The re-emergence of emergence. In: Chap. Strong and weak emergence. Oxford University Press, London (2006)
6. Conrad, E.D., Tyson, J.J.: System modelling in cellular biology: from concepts to nuts and bolt. In: Chap. Modelling molecular interaction networks with nonlinear ordinary differential equations, pp. 97–125. MIT, MA (2006)
7. Corning, P.A.: Holistic Darwinism: synergy, cybernetics, and the bioeconomics of evolution. University of Chicago Press, IL (2005)
8. Dawkins, R.: Climbing mount improbable (1997)
9. Faisal, A.A., Selen, L.P.J., Wolpert, D.M.: Noise in the nervous system. Nat Rev Neurosci **9**(4), 292–303 (2008). doi:10.1038/nrn2258. http://dx.doi.org/10.1038/nrn2258
10. Fogel, D.B.: Evolutionary computation: toward a new philosophy of machine intelligence, 3rd edn. Wiley, NY (2006)
11. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J Phys Chem **81**(25), 2340–2361 (1977). http://dx.doi.org/10.1021/j100540a008
12. Gillespie, D.T., Petzold, L.R.: System modeling in cellular biology, from concepts to nuts and bolts. In: Chap. Numerical simulation for biochemical kinetics, pp. 331–353. MIT, MA (2006)
13. Hoffmeister, F., Back, T.: Genetic algorithms and evolution strategies: similarities and differences. In: Lecture notes in computer science, vol. 496, pp. 455–469. Springer, Berlin (1991)
14. Kitano, H.: Systems biology: a brief overview. Science **295**(5560), 1662–1664 (2002)
15. Klamt, S., Stelling, J.: System modelling in cellular biology: from concepts to nuts and bolt. In: Chap. Stoichiometric and constraint-based modeling, pp. 73–96. MIT, MA (2006)
16. Leloup, J.C., Goldbeter, A.: Modeling the circadian clock: from molecular mechanism to physiological disorders. Bioessays **30**(6), 590–600 (2008). doi:10.1002/bies.20762. http://dx.doi.org/10.1002/bies.20762
17. Liebermeister, W., Klipp, E.: Biochemical networks with uncertain parameters. In: Systems Biology, IEE Proceedings, vol. 152, pp. 97–107 (2005)
18. Moles, C.G., Mendes, P., Banga, J.R.: Parameter estimation in biochemical pathways: a comparison of global optimization methods. Genome Res **13**(11), 2467–2474 (2003). doi:10.1101/gr.1262503. http://dx.doi.org/10.1101/gr.1262503

19. Mosca, E., Merelli, I., Alfieri, R., Milanesi, L.: A distributed approach for parameter estimation in systems biology models. Il Nuovo Cimento C **2**, 165–168 (2009)
20. Novre, N.L., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., Bergman, F.T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villger, A., Boyd, S.E., Calzone, L., Courtot, M., Dogrusoz, U., Freeman, T.C., Funahashi, A., Ghosh, S., Jouraku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G., Goryanin, I., Kell, D.B., Sander, C., Sauro, H., Snoep, J.L., Kohn, K., Kitano, H.: The systems biology graphical notation. Nat Biotechnol **27**(8), 735–741 (2009). doi:10.1038/nbt.1558. http://dx.doi.org/10.1038/nbt.1558
21. Runarsson, T.P., Yao, X.: Stochastic ranking for constrined evolutionary optimization. IEEE Trans Evol Optim **4**(3), 284–294 (2000)
22. Saravanan, N., Fogel, D.B., Nelson, K.M.: A comparison of methods for self-adaptation in evolutionary algorithms. Biosystems **36**(2), 157–166 (1995)
23. Sole, R., Goodwin, B.: Signs of life. How complexity pervades biology. BasicBooks, NY (2000)
24. Stelling, J., Sauer, U., III, F.J.D., Doyle, J.: System modeling in cellular biology: from concepts to nuts and bolts. In: Chap. Complexity and robustenss of cellular network, pp. 19–41. MIT, MA (2006)
25. Tyson, J.J., Chen, K.C., Novak, B.: Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. Curr Opin Cell Biol **15**(2), 221–231 (2003)
26. Vera, J., Balsa-Canto, E., Wellstead, P., Banga, J.R., Wolkenhauer, O.: Power-law models of signal transduction pathways. Cell Signal **19**(7), 1531–1541 (2007). doi:10.1016/j.cellsig.2007.01.029. http://dx.doi.org/10.1016/j.cellsig.2007.01.029
27. Westerhoff, H.V., Palsson, B.O.: The evolution of molecular biology into systems biology. Nat Biotechnol **22**(10), 1249–1252 (2004)

Chapter 7
Haplotype Inference Using Propositional Satisfiability

Ana Graça, João Marques-Silva, and Inês Lynce

Abstract Haplotype inference is an important problem in computational biology, which has deserved large effort and attention in the recent years. Haplotypes encode the genetic data of an individual at a single chromosome. However, humans are diploid (chromosomes have maternal and paternal origin), and it is technologically infeasible to separate the information from homologous chromosomes. Hence, mathematical methods are required to solve the haplotype inference problem. A relevant approach is the pure parsimony. The haplotype inference by pure parsimony (HIPP) aims at finding the minimum number of haplotypes which explains a given set of genotypes. This problem is NP-hard.

Boolean satisfiability (SAT) has successful applications in several fields. The use of SAT-based techniques with pure parsimony haplotyping has shown to produce very efficient results. This chapter describes the haplotype inference problem and the SAT-based models developed to solve the problem. Experimental results confirm that the SAT-based methods represent the state of the art in the field of HIPP.

7.1 Introduction

Recent advances in sequencing technologies have enabled sequencing the genome of thousands of people efficiently and inexpensively. Such information has offered investigators new opportunities to understand the genetic differences between human beings, and later mapping such differences with common human diseases.

The International HapMap Project[1] and the 1000 Genomes Project[2] represent significant efforts to catalog the genetic variations among human beings.

[1] http://www.hapmap.org

[2] http://www.1000genomes.org

A. Graça (✉)
IST/INESC-ID, Technical University of Lisbon, R. Alves Redol 9, 1000-029 Lisboa, Portugal
e-mail: assg@sat.inesc-id.pt

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_7,

At the forehead of human variation at genetic level are single nucleotide polymorphisms (SNPs). An SNP is a single DNA position where a mutation has occurred and one nucleotide was substituted with a different one. Moreover, the least frequent nucleotide must be present in a significant percentage of the population (e.g., 1%). SNPs are the most common genetic variation. The human genome has millions of SNPs [42], which are cataloged in dbSNP,[3] the public repository for DNA variations [40].

Haplotypes correspond to the sequence of SNPs in a single chromosome which are inherited together. Humans are diploid organisms, which mean that our genome is organized in pairs of homologous chromosomes, representing the maternal and paternal chromosome. Therefore, each individual has two haplotypes for a given stretch of the genome. Genotypes correspond to the conflated data of homologous haplotypes.

Technological limitations prevent geneticists from acquiring experimentally the data from a single chromosome, the haplotypes. Instead, genotypes are obtained. This means that at each DNA position it is possible to know whether the individual has inherited the same nucleotide from both parents (homozygous positions) or distinct nucleotides from each parent (heterozygous positions). Nonetheless, in the latter case, it is, in general, technologically infeasible to determine which nucleotide was inherited from each parent. The problem of obtaining the haplotypes from the genotypes is known as haplotype inference.

Information about human's haplotypes has significant importance in clinic medicine [8]. Haplotypes are more informative than genotypes and, in some cases, can predict better the severity of a disease or even be responsible for producing a specific phenotype. In some cases of medical transplants, patients who match the donor haplotypes closely are predicted to have more success on the transplant outcome [35]. Moreover, medical treatments could be customized based on patient's genetic information, because individual responses to drugs can be attributed to a specific haplotype [15]. Furthermore, haplotypes can help inferring population histories.

Despite being an important biological problem, haplotype inference turned also to be a challenging mathematical problem and, therefore, has deserved significant attention by the mathematical and computer science communities. The mathematical approaches to haplotype inference can be statistical [4, 41] or combinatorial [6, 18, 19]. Within the combinatorial methods, the haplotype inference by pure parsimony (HIPP) approach [19] is noteworthy. The pure parsimony approach aims at finding the haplotype inference solution which uses a smaller number of haplotypes. The HIPP problem is APX-hard [28].

Boolean satisfiability (SAT) has been successfully applied in a significant number of different fields [33]. The application of SAT-based methodologies in haplotype inference has been shown to produce very competitive results when compared to alternative methods [17, 31]. SAT-based models currently represent the state of the art on HIPP and, therefore, are the main focus of this chapter.

[3] http://www.ncbi.nlm.nih.gov/projects/SNP

This chapter starts by describing the haplotype inference problem, with special focus on the pure parsimony approach. Follows an overview of the mathematical models suggested for solving the problem. Later, the SAT-based haplotype inference model and the model's extension to handle polyploid species are detailed. In addition, the pseudo-Boolean optimization (PBO) model and its extension to deal with data with missing sites are presented. Moreover, this chapter summarizes an experimental evaluation involving a considerable number of maximum parsimony haplotyping algorithms. Furthermore, standard preprocessing techniques commonly used by HIPP algorithms, which include structural simplifications on genotype instances and calculation of bounds, are described.

7.2 Haplotype Inference

The genome constitutes the hereditary data of an organism and is encoded in the DNA (*deoxyribonucleic acid*), which is specified by the sequence of bases of nucleotides that represent the DNA structural units: A (*adenine*), C (*cytosine*), T (*thymine*) and G (*guanine*).

The coding part of the genome is organized in DNA segments called genes. Each gene encodes a specific protein. The variants of a single gene are named *alleles*. Despite the considerable similarity between our genes, no two individuals have the same genome. The human genome has roughly three billion nucleotides, but about 99.9% of them are the same for all human beings. On average, the sequence of bases of two individuals differ in one of every 1,200 bases, but the variations are not uniformly distributed along all the DNA. Variations in the DNA define the differences between human beings and, in particular, influence their susceptibility to diseases. Consequently, a critical step in genetics is the understanding of the differences between human beings. SNPs correspond to differences in a single position of the DNA where mutations have occurred and present a minor allele frequency equal to or greater than a given value (e.g., 1%).

SNPs which are close on the genome tend to be inherited together in blocks. Hence, SNPs within a block are statistically associated, what is known as linkage disequilibrium. These blocks of SNPs are known as haplotypes. *Haplotypes* are therefore sequences of correlated SNPs (Fig. 7.1). Haplotype blocks exist because the crossing-over phenomenon (exchange of genetic material between homologous chromosomes during meiosis) does not occur randomly along the DNA, but it is rather concentrated into small regions called recombination hotspots. Recombination does not occur in every hotspot at every generation. Consequently, individuals within the same population tend to have large haplotype blocks in common. Furthermore, due to the association of SNPs, it is often possible to identify a small subset of SNPs which identify the remaining SNPs within the haplotype (*tagSNPs*) [24].

The human genome is organized into 22 pairs of homologous non-sex chromosomes, each chromosome being inherited from one parent. Due to technological limitations, homologous chromosomes are not easy to sequence separately, and

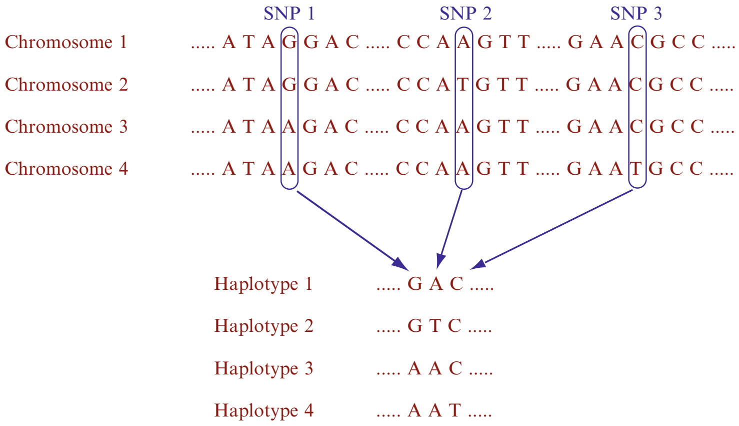

SNP 1
SNP 2
SNP 3
Chromosome 1 ..... A T A G G A C ..... C C A A G T T ..... G A A C G C C .....
Chromosome 2 ..... A T A G G A C ..... C C A T G T T ..... G A A C G C C .....
Chromosome 3 ..... A T A A G A C ..... C C A A G T T ..... G A A C G C C .....
Chromosome 4 ..... A T A A G A C ..... C C A A G T T ..... G A A T G C C .....
Haplotype 1 ..... G A C .....
Haplotype 2 ..... G T C .....
Haplotype 3 ..... A A C .....
Haplotype 4 ..... A A T .....

Fig. 7.1 Identifying SNPs and haplotypes

consequently, it is not possible to obtain the haplotypes which correspond to a single chromosome. Instead, genotypes, which correspond to the conflated data of two haplotypes on homologous chromosomes, are obtained. In general, the genotype data do not make it possible to distinguish between the alleles inherited from each of the parents. The haplotype inference problem corresponds to finding the set of haplotypes which originate from a given set of genotypes.

Almost all human SNPs are *biallelic*, which means that only two different alleles are allowed for that position. SNPs with more than two different alleles are called *polyallelic*. In what follows we will only consider biallelic SNPs, with two possible alleles. The *wild type* corresponds to the more common allele, and the *mutant type* corresponds to the less frequent allele.

A *haplotype* is the genetic constitution of an individual chromosome. The underlying data that form a haplotype can be the full DNA sequence in the region, or more commonly the SNPs in that region. Diploid organisms pair homologous chromosomes, thus containing two haplotypes, one inherited from each parent. The *genotype* describes the conflated data of the two haplotypes. In other words, an *explanation* for a genotype is a pair of haplotypes. Conversely, this pair of haplotypes explains the genotype. If for a given site both copies of the haplotype have the same value, then the genotype is said to be *homozygous* at that site; otherwise it is said to be *heterozygous*.

Despite the biological origin of the haplotype inference problem, it can be described as a mathematical problem. Given a set $\mathcal{G}$ of n genotypes g_i, with $1 \leq i \leq n$, i.e., sequences of length m, the haplotype inference problem consists in finding a set $\mathcal{H}$ of haplotypes, such that for each genotype $g_i \in \mathcal{G}$ there is at least one pair of haplotypes (h_j, h_k), with h_j and $h_k \in \mathcal{H}$ such that the pair (h_j, h_k) explains g_i. The value n denotes the number of individuals in the sample, and m denotes the number of SNP sites. Furthermore, g_{ij} denotes a specific site j in genotype g_i, with $1 \leq j \leq m$. Without loss of generality, we may assume that the values of a SNP are always 0 or 1. Value 0 represents the wild type and value 1 represents the mutant. A haplotype is then a string over the alphabet {0,1}. Moreover, genotypes may be represented by extending the alphabet used for representing haplotypes to {0,1,2}. Homozygous sites are then represented by values 0 or 1, depending on whether both haplotypes have value 0 or 1 at that site, respectively. Heterozygous sites are represented by value 2. A genotype g_i is explained by a pair (h_i^a, h_i^b) of haplotypes. This fact is represented by $g_i = h_i^a \oplus h_i^b$ with

$$g_{ij} = \begin{cases} 0 \text{ if } h_{ij}^a = h_{ij}^b = 0 \\ 1 \text{ if } h_{ij}^a = h_{ij}^b = 1 \\ 2 \text{ if } h_{ij}^a \neq h_{ij}^b \end{cases},$$

for each specific site g_{ij}, with $1 \leq j \leq m$.

Definition 7.1. (Haplotype Inference) Given a set with n genotypes, $\mathscr{G}$, each with size m, the haplotype inference problem aims at finding the set of haplotypes, $\mathscr{H}$, which originate the genotypes in $\mathscr{G}$ and associating a pair of haplotypes (h_i^a, h_i^b), with $h_i^a, h_i^b \in \mathscr{H}$, to each genotype $g_i \in \mathscr{G}$, such that $g_i = h_i^a \oplus h_i^b$.

Example 7.1. (Haplotype Inference) Consider genotype 02212 having five sites, of which one SNP is homozygous with value 0, one SNP is homozygous with value 1 and the remaining three SNPs correspond to heterozygous sites. There are four different possible explanations for this genotype: (00010, 01111), (00110, 01011), (00111, 01010) and (00011, 01110).

For each genotype $g_i \in \mathscr{G}$ with z heterozygous positions, there are 2^{z-1} possible pairs of haplotypes which can explain g_i. Choosing the biological correct haplotype pair would be impossible without the implicit or explicit use of some genetic model to guide the algorithm in constructing a solution. The coalescent model [22] states that there is a unique ancestor for all individuals of the same population. In this chapter, we consider the pure parsimony approach which is indirectly related to the coalescent genetic model.

7.2.1 *Haplotype Inference by Pure Parsimony*

The most explored combinatorial approach to the haplotype inference problem is called HIPP [19]. A solution to this problem minimizes the total number of distinct haplotypes being used. The idea of searching for the solution with the smallest number of haplotypes is biologically motivated by the fact that individuals from the same population have the same ancestors and mutations do not occur often. Moreover, empirical results provide support for this method: the number of haplotypes in a large population is typically very small, although genotypes exhibit a great diversity.

Definition 7.2. The HIPP problem consists in finding a minimum-size set $\mathscr{H}$ of haplotypes that explain all genotypes in $\mathscr{G}$.

Example 7.2. (HIPP) Consider the set of genotypes $\mathscr{G} = \{g_1, g_2, g_3\} = \{022, 221, 222\}$. There are solutions using six different haplotypes: $\mathscr{H}_1 = \{000, 001, 010, 011, 101, 111\}$, such that $g_1 = 001 \oplus 010$, $g_2 = 011 \oplus 101$ and $g_3 = 000 \oplus 111$. However, the HIPP solution only requires four distinct haplotypes: $\mathscr{H}_2 = \{000, 011, 101, 111\}$ such that $g_1 = 011 \oplus 000$, $g_2 = 011 \oplus 101$ and $g_3 = 011 \oplus 100$.

In general, there may exist more than one solution to the problem.

Finding a solution to the HIPP problem is an APX-hard (and consequently, NP-hard) problem [28]. This means that not only there is no polynomial time algorithm to solve the problem, but also does not exist a polynomial time approximation scheme, unless P = NP.

7.3 Related Work

A significant number of methods have been developed to solve the pure parsimony haplotyping problem, pursuing the efficiency goal.

Different constraint techniques have been applied to solving the HIPP problem. The most used technique is integer linear programming (ILP) [1–3, 5, 19, 20, 28]. Other techniques are based on branch-and-bound [27, 44], Boolean satisfiability (SAT) [29, 36], pseudo-Boolean optimization (PBO) [16], answer set programming (ASP) [12] and constraint programming (CP) [37].

The first HIPP model was proposed by Gusfield [19] and is an integer linear programing approach. The main drawback of Gusfield's model, RTIP, is its size because the model has an exponential number of variables and constraints. RTIP considers enumerating all pairs of haplotypes which explain every genotype. Given that, for each genotype $g \in \mathscr{G}$, the number of haplotype pairs which explain g is 2^z, where z is the number of heterozygous positions of g, RTIP turns out to be an exponential model. Nonetheless, the model is simplified by not considering haplotype pairs where both haplotypes cannot explain more than one genotype. Note that these haplotype pairs can be removed from the formulation while maintaining the correctness of the algorithm. This simplification makes the model practical in several situations, specially when the level of recombination is high, because there exists more haplotype diversity. Even though, the model still has its exponentially as a drawback.

The RTIP model inspired a branch-and-bound algorithm to the HIPP problem, known as HAPAR [44]. A more recent branch-and-bound algorithm is used to solve an ILP model based on a set-covering formulation of the problem [27].

PolyIP [2] is a model proposed by Brown and Harrower, which is polynomial on the number of genotypes and sites. Similar formulations were independently suggested by other authors [20, 28]. PolyIP associates two haplotypes with each genotype. A Boolean variable is associated with each haplotype site and constraints ensure that each haplotype pair explains the corresponding genotype. Moreover, for each two haplotypes, the model needs variables defined as true if the haplotypes are different. Furthermore, for each haplotype there exists a variable which is true if the haplotype is different from all previous haplotypes. The cost function aims at minimizing those variables, thus minimizing the number of different haplotypes. The number of constraints and variables of the PolyIP model are, respectively, in $\Theta(n^2 m)$ and $\Theta(n^2 + nm)$, where n represents the number of genotypes and m represents the number of sites.

The same authors of PolyIP also proposed another ILP model, HybridIP [3], which represents a hybrid between the RTIP and the PolyIP models. HybridIP was formulated to combine the strengths of RTIP and PolyIP, in an approach with practical size and able to run within reasonable run times. Nonetheless, no practical significant improvements were achieved by this new model compared with PolyIP, as confirmed by experimental results (Sect. 7.6).

More recently, a distinct polynomial ILP formulation, based on class representatives, was proposed, HaploPPH [5]. The idea is that a solution for HIPP induces a covering of the genotypes by subsets such that each subset of genotypes share one haplotype, each genotype belongs to exactly two subsets and every pair of genotypes intersects in at the most one genotype.

HAPLO-ASP uses answer set programming (ASP) to solve the HIPP problem. ASP is a declarative programming paradigm which represents the computational problem as a program whose models correspond to solutions of the problem. The haplotype inference solution is computed using an answer set solver named Cmodels [14] and uses a SAT solver as a search engine, MiniSat [11].

An alternative solution which uses constraint programming for solving HIPP is reported in [37], but as confirmed by the author, it is not as efficient as the SAT-based models.

In addition, a significant number of polynomial heuristic approaches have been proposed [13,21,28,43,45]. However, given that the problem is APX-hard, no polynomial heuristic algorithm can guarantee an approximation algorithm.

7.4 SAT-Based Haplotype Inference

The original SAT-based model is called SHIPs [29] and represents a notable improvement in the efficiency of existing HIPP solvers. In addition, the SAT-based approach has been generalized to polyploid and polyallelic data [36]. Furthermore, the haplotype inference problem has been successfully tackled using PBO [16], which is a generalization of SAT capable of handling optimization functions.

7.4.1 Background on Boolean Satisfiability

In Boolean logic (or propositional logic), each variable x_i may take two values, 1 (for true) or 0 (for false). A *literal* l_i is a variable x_i or its negation $\neg x_i$. A *clause* is a disjunction ($\vee$) of literals. A formula, φ, in CNF (conjunctive normal form) is the conjunction ($\wedge$) of clauses. For example, $\varphi = (x_1 \wedge x_2) \vee (\neg x_1 \wedge x_3) \vee (\neg x_2 \wedge \neg x_3)$ is a CNF formula with three variables and three clauses. A CNF formula, φ, is satisfied if all clauses in φ are satisfied. Every propositional formula can be converted into an equivalent formula which is in CNF.

Definition 7.3. The SAT problem (or *propositional satisfiability* problem) aims at deciding whether there exists a Boolean assignment to the variables in a given Boolean CNF formula, φ, such that φ becomes satisfied and, if that is the case, provides a satisfying assignment.

For example, a possible SAT solution to the formula φ given above assigns $x_1 = 1$, $x_2 = 0$ and $x_3 = 1$.

The SAT problem was the first to be proved NP-complete, in 1971 [7].

A PBO problem is a generalization of SAT, when the Boolean formula is extended with an optimization function. A PBO problem is described by a set of a pseudo-Boolean formulas and an optimization function.

7.4.2 The SHIPs Model

This SAT-based formulation models whether there exists a set $\mathscr{H}$ of haplotypes, with $r = |\mathscr{H}|$ haplotypes, such that each genotype $g_i \in \mathscr{G}$ is explained by a pair of haplotypes in $\mathscr{H}$. The SAT-based algorithm considers increasing sizes for $\mathscr{H}$, from a lower bound lb to an upper bound ub. Trivial lower and upper bounds are, respectively, 1 and $2 \cdot n$. The algorithm terminates for a size of $\mathscr{H}$ for which there exists $r = |\mathscr{H}|$ haplotypes such that every genotype in $\mathscr{G}$ is explained by a pair of haplotypes in $\mathscr{H}$. In what follows we assume n genotypes each with m sites. The same indexes will be used throughout: i ranges over the genotypes and j over the sites, with $1 \leq i \leq n$ and $1 \leq j \leq m$. In addition, r candidate haplotypes are considered, each with m sites. An additional index k is associated with haplotypes, $1 \leq k \leq r$. As a result, $h_{kj} \in \{0, 1\}$ denotes the jth site of haplotype k. Moreover, a haplotype h_k is viewed as a m-bit word, $h_{k1} \dots h_{km}$. A valuation $v : \{h_{k1}, \dots, h_{km}\} \to \{0, 1\}$ to the bits of h_k is denoted by h_k^v. Observe that valuations can be extended to other sets of variables.

For a given value of r, the model considers r haplotypes and seeks to associate two haplotypes (which can possibly represent the same haplotype) with each genotype g_i, $1 \leq i \leq n$. As a result, for each genotype g_i, the model uses *selector* variables for selecting which haplotypes are used for explaining g_i. Since each genotype is to be explained by *two* haplotypes, the model uses two sets, a and b, of r selector variables, respectively s_{ki}^a and s_{ki}^b, with $k = 1, \dots, r$. Hence, genotype g_i is explained by haplotypes h_{k_1} and h_{k_2} if $s_{k_1 i}^a = 1$ and $s_{k_2 i}^b = 1$. Clearly, g_i is also explained by the same haplotypes if $s_{k_2 i}^a = 1$ and $s_{k_1 i}^b = 1$.

If a site g_{ij} of a genotype g_i is either 0 or 1, then this is the value required at this site and so this information is used by the model. If a site g_{ij} is 0, then the model requires, for $k = 1, \dots, r$,

$$\left(\neg h_{kj} \vee \neg s_{ki}^{a}\right) \wedge \left(\neg h_{kj} \vee \neg s_{ki}^{b}\right). \tag{7.1}$$

If a site g_{ij} is 1, then the model requires, for $k = 1, \ldots, r$,

$$\left(h_{kj} \vee \neg s_{ki}^{a}\right) \wedge \left(h_{kj} \vee \neg s_{ki}^{b}\right). \tag{7.2}$$

Otherwise, one requires that the haplotypes explaining genotype g_i have opposite values at site i. This is done by creating two variables, $g_{ij}^{a} \in \{0, 1\}$ and $g_{ij}^{b} \in \{0, 1\}$, such that $g_{ij}^{a} \neq g_{ij}^{b}$. In CNF, the model requires two clauses,

$$\left(g_{ij}^{a} \vee g_{ij}^{b}\right) \wedge \left(\neg g_{ij}^{a} \vee \neg g_{ij}^{b}\right). \tag{7.3}$$

In addition, the model requires, for $k = 1, \ldots, r$,

$$\begin{aligned} &\left(h_{kj} \vee \neg g_{ij}^{a} \vee \neg s_{ki}^{a}\right) \wedge \left(\neg h_{kj} \vee g_{ij}^{a} \vee \neg s_{ki}^{a}\right) \\ \wedge &\left(h_{kj} \vee \neg g_{ij}^{b} \vee \neg s_{ki}^{b}\right) \wedge \left(\neg h_{kj} \vee g_{ij}^{b} \vee \neg s_{ki}^{b}\right). \end{aligned} \tag{7.4}$$

Clearly, for each value of i, and for a and b, it is necessary that exactly one haplotype is used, and so exactly one selector variable be assigned value 1. This can be captured with cardinality constraints,

$$\left(\sum_{k=1}^{r} s_{ki}^{a} = 1\right) \wedge \left(\sum_{k=1}^{r} s_{ki}^{b} = 1\right). \tag{7.5}$$

The SHIPs model can also be described using a matrix formulation $G = S^a \cdot H \oplus S^b \cdot H$, where G is a $n \times m$ matrix describing the genotypes, H is a $r \times m$ matrix of haplotype variables, S^a and S^b are the $n \times r$ matrices of selector variables and $\oplus$ is the explanation operation.

Theorem 7.1. *(Space Complexity) If a solution is found with r_f haplotypes, then the number of constraints of the SAT model is $\mathcal{O}(r_f\, nm)$, which is $\mathcal{O}(n^2 m)$. In addition, the number of variables is $\mathcal{O}(nm + r_f m + r_f n)$, which is $\mathcal{O}(n^2 + nm)$.*

The core SHIPs model is not effective in practice. As a result, several key improvements have been developed, which are essential for obtaining significant performance gains over existing approaches.

A crucial technique is the utilization of a tight lower bound estimate, which reduces the number of iterations of the algorithm, but also effectively prunes the search space. The computation of lower bounds is discussed in Sect. 7.5.2.

One additional key technique for pruning the search space is motivated by observing the existence of symmetry in the problem formulation. Consider two haplotypes h_{k_1} and h_{k_2}, and the selector variables $s_{k_1 i}^{a}$, $s_{k_2 i}^{a}$, $s_{k_1 i}^{b}$ and $s_{k_2 i}^{b}$. Furthermore, consider Boolean valuations v_x and v_y to the sites of haplotypes h_{k_1} and h_{k_2}. Then, $h_{k_1}^{v_x}$ and $h_{k_2}^{v_y}$, with $s_{k_1 i}^{a} s_{k_2 i}^{a} s_{k_1 i}^{b} s_{k_2 i}^{b} = 1001$, corresponds to $h_{k_1}^{v_y}$ and $h_{k_2}^{v_x}$, with $s^a_{k_1\,i}s^a_{k_2\,i}s^b_{k_1\,i}s^b_{k_2\,i} = 0110$, and one of the assignments can be eliminated. To remedy this, one possibility is to enforce an ordering of the Boolean valuations to the haplotypes. Hence, for any valuation v to the problem variables, we require $h^v_1 < h^v_2 < \ldots < h^v_r$.

7.4.2.1 Improvements to SHIPs

Motivated by an effort to apply the SHIPs model to biological test data, we were able to identify a number of additional improvements to the basic model. For difficult problem instances, the run time is very sensitive to the number of g variables used. The basic model creates two variables for each heterozygous site. One simple optimization is to replace each pair of g variables associated with a heterozygous site, g^a_{ij} and g^b_{ij}, by a single variable t_{ij}. Consequently, the new set of constraints becomes:

$$\begin{aligned}(h_{kj} \vee \neg t_{ij} \vee \neg s^a_{ki}) \wedge (\neg h_{kj} \vee t_{ij} \vee \neg s^a_{ki}) \wedge \\ (h_{kj} \vee t_{ij} \vee \neg s^b_{ki}) \wedge (\neg h_{kj} \vee \neg t_{ij} \vee \neg s^b_{ki}).\end{aligned} \tag{7.6}$$

Hence, if selector variable s^a_{ki} is activated (i.e., assumes value 1), then h_{kj} is equal to t_{ij}. In contrast, if selector variable s^b_{ki} is activated, then h_{kj} is the complement of t_{ij}. Observe that, since the genotype has at least one heterozygous site, then it must be explained by *two different* haplotypes, and so s^a_{ki} and s^b_{ki} cannot be simultaneously activated.

The basic model uses lower bounds, which are obtained by identifying incompatibility relations among genotypes. Two genotypes are *incompatible* if there exists a site for which the value of one genotype is 0 and the value of the other genotype is 1. Otherwise, the genotypes are *compatible*.These incompatibility relations find other applications. Consider two incompatible genotypes, g_{i_1} and g_{i_2}, and a candidate haplotype h_k. Hence, if either $s^a_{ki_1}$ or $s^b_{ki_1}$ is activated, and so h_k is used for explaining genotype g_{i_1}, then haplotype h_k *cannot* be used for explaining g_{i_2}; hence both $s^a_{ki_2}$ and $s^b_{ki_2}$ *must not* be activated. The implementation of this condition is achieved by adding the following clauses for each pair of incompatible genotypes g_{i_1} and g_{i_2} and for each candidate haplotype h_k,

$$\left(\neg s^a_{ki_1} \vee \neg s^a_{ki_2}\right) \wedge \left(\neg s^a_{ki_1} \vee \neg s^b_{ki_2}\right) \wedge \left(\neg s^b_{ki_1} \vee \neg s^a_{ki_2}\right) \wedge \left(\neg s^b_{ki_1} \vee \neg s^b_{ki_2}\right). \tag{7.7}$$

One of the key techniques proposed in the basic model is the utilization of the sorting condition over the haplotypes, as an effective symmetry breaking technique. Additional symmetry breaking conditions are possible. Note that the model consists of selecting a candidate haplotype for the a representative and another haplotype for the b representative, such that each genotype is explained by the a and b representatives. Given a set of r candidate haplotypes, let h_{k_1} and h_{k_2}, with $k_1, k_2 \leq r$, be two haplotypes which explain a genotype g_i. This means that g_i can be explained not only by the assignments $s^a_{k_1\,i}s^a_{k_2\,i}s^b_{k_1\,i}s^b_{k_2\,i} = 1001$ but also by the assignments

$s^a_{k_1 i} s^a_{k_2 i} s^b_{k_1 i} s^b_{k_2 i} = 0110$. This symmetry can be eliminated requiring only one arrangement of the s variables to be used to explain each genotype g_i. One solution is to require the haplotype selected by the s^a_{ki} variables to have an index *smaller* than the haplotype selected by the s^b_{ki} variables. This requirement is captured by the conditions:

$$\left(s^a_{k\,i} \Rightarrow \bigwedge_{k_2=1}^{k-1} \neg s^b_{k_2 i} \right) \text{ and } \left(s^b_{k\,i} \Rightarrow \bigwedge_{k_1=k+1}^{r} \neg s^a_{k_1 i} \right). \tag{7.8}$$

Clearly, each condition above can be represented by a single clause, for each k_1 (or k_2) and i. Moreover, note that for genotypes with heterozygous sites, the upper limit of the first constraint can be set to k and the lower limit of the second condition can be set to k.

7.4.3 The SATlotyper Model

The majority of the haplotype inference methods can only handle biallelic SNP data of diploid species. Indeed, human beings and most animals are diploid species, i.e., with two homologous sets of chromosomes. However, polyploidy, i.e., more than two sets of homologous chromosomes, is common in plants. Moreover, although the large majority of SNPs are biallelic, there are exceptions with three or four possible alleles.

The SATlotyper method [36] is a relevant contribution which corresponds to a generalization of the SAT-based approach to handle polyallelic data and polyploid species. Therefore, the constraints generated by SATlotyper are extensions of the constraints generated by SHIPs.

This section presents the SAT model for biallelic polyploids. Although SATlotyper is also able to handle SNPs with more than two possible values, for that case we refer to the original paper [36].

Let p be the ploidy of the considered species, i.e., the species has p-tuples of homologous chromosomes. Then each polyploid genotype g_i, with $1 \leq i \leq n$ is represented by a sequence of m vectors with size p, where each vector encodes one SNP site of the given individual, $g_{ij} = (g^1_{ij}, g^2_{ij}, \ldots, g^p_{ij})$, with $1 \leq j \leq m$. A site g_{ij} is heterozygous if there are two components, $g^{l_1}_{ij}$ and $g^{l_2}_{ij}$, such that $g^{l_1}_{ij} \neq g^{l_2}_{ij}$, with $1 \leq l_1, l_2 \leq p$. The haplotype inference problem must be reformulated.

Definition 7.4. (Haplotype Inference – Polyploid Species) Given a set $\mathscr{G}$ with n genotypes, each of length m, the haplotype inference problem aims at finding a set of haplotypes $\mathscr{H}$, such that for each genotype $g_i \in \mathscr{G}$, there exists a non-ordered tuple of p haplotypes $(h^1_i, \ldots, h^p_i)$, with $h^1_i, \ldots, h^p_i$ explaining genotype g_i.

A set with p haplotypes explains a genotype g_i if the p haplotypes and the genotype g_i have the same allele composition at each SNP site.

Example 7.3. (Haplotype Inference – Polyploid Species) An example of a tetraploid genotype with three biallelic SNP sites is $g_i = (1,0,0,1)(0,0,0,1)(1,1,1,1)$. This genotype can have 24 possible explanations, corresponding to all possible permutations of alleles at each g_{ij} position.

The core algorithm of SATlotyper is the same as in the basic SHIPs model. The model is defined iteratively from a lower bound to an upper bound. Trivial lower and upper bounds are, respectively, 1 and $p \cdot n$. As in SHIPs, the model uses selector variables for selecting which haplotypes are used for explaining each genotype. For each genotype, the model uses p sets of selector variables, $s^l_{k\,i}$, for $1 \leq l \leq p$. Haplotypes $h_{k_1}, \ldots, h_{k_p}$ are selected to explain g_i if $s^l_{k_1\,i} = 1, \ldots, s^l_{k_p\,i} = 1$.

For each genotype site g_{ij}, the selector constraints are described as follows. If $g^l_{ij} = 0$, for all $1 \leq l \leq p$, then every haplotype which is chosen to explain g_i must have 0 at site j,

$$\left(\neg h_{kj} \vee \neg s^l_{ki}\right), \tag{7.9}$$

with $1 \leq k \leq r$. If $g^l_{ij} = 1$, for all $1 \leq l \leq p$, then every haplotype which is chosen to explain g_i must have 1 at site j,

$$\left(h_{kj} \vee \neg s^l_{ki}\right), \tag{7.10}$$

with $1 \leq k \leq r$. Otherwise, if the genotype site g_{ij} is heterozygous, it is necessary to create p Boolean variables $g^1_{ij}, \ldots, g^p_{ij}$, which represent the possible arrangements of 1s and 0s at site j.

$$\left(h_{kj} \vee \neg g^l_{ij} \vee \neg s^l_{ki}\right) \wedge \left(\neg h_{kj} \vee g^l_{ij} \vee \neg s^l_{ki}\right), \tag{7.11}$$

where $1 \leq k \leq r$ and $1 \leq l \leq p$. Finally, for each i and l, it is necessary that exactly one haplotype is selected. This is represented by the cardinality constraint

$$\sum_{k=1}^{r} s^l_{k\,i} = 1. \tag{7.12}$$

Furthermore, the model applies symmetry breaking to haplotypes and genotypes, similarly to SHIPs.

The number of variables is $\mathcal{O}(nmp^2 \log_2 p)$ and the number of constraints is $\mathcal{O}(r_f nmp)$, where r_f is the final (most parsimonious) number of haplotypes.

7.4.4 The RPoly Model

The success of solving the HIPP problem using SAT techniques motivated the consideration of other SAT-based procedures, such as PBO methods. This section describes the PBO approach for solving the HIPP problem: the RPoly model.

The RPoly model is a notable improvement over the PolyIP model. A significant number of modifications which are shown to be very effective are proposed: first, the use of a PBO solver instead of an ILP solver; second, symmetry breaking and reduction of the size of the model; finally, the integration of lower bounds and cardinality constraints.This section describes these RPoly features and, moreover, extends the RPoly model to include genotypes with missing sites.

The RPoly model associates two haplotypes (h_i^a, h_i^b) with each genotype $g_i \in \mathcal{G}$, and conditions are defined which capture when a different haplotype is used for explaining a given genotype.

RPoly associates variables only with heterozygous sites. Note that homozygous sites do not require variables because the value of the haplotypes explaining homozygous sites is known beforehand and so can be implicitly assumed. Therefore, a Boolean variable t_{ij} is associated with each heterozygous site g_{ij}, such that,

$$t_{ij} = \begin{cases} 1 \text{ if } h_{ij}^a = 1 \wedge h_{ij}^b = 0 \\ 0 \text{ if } h_{ij}^a = 0 \wedge h_{ij}^b = 1. \end{cases} \tag{7.13}$$

This alternative definition of the variables associated with the sites of genotypes reduces the number of variables by a factor of 2. In addition, the model only creates variables for heterozygous sites, and therefore the number of variables associated with sites equals the total number of heterozygous sites. It should be mentioned that this definition of the variables associated with sites follows the SHIPs model [29,30].

Hence, the existing symmetry in haplotype pairs described in SHIPs model is broken by considering that $t_{ij} = 0$ for each first heterozygous site g_{ij} of each genotype g_i,

$$\left(g_{ij} = 2 \wedge \forall_{\hat{j}} \left(\hat{j} < j \Rightarrow g_{i\hat{j}} \neq 2\right)\right) \Longrightarrow \neg t_{ij}. \tag{7.14}$$

Candidate haplotypes for each genotype are related to candidate haplotypes for other genotypes only if the two genotypes are compatible. Clearly, incompatible genotypes cannot be explained by common haplotypes. In practice, for candidate haplotypes h_i^p and h_k^q $(p, q \in \{a, b\}$ and $1 \leq k < i \leq n)$, a Boolean variable x_{ik}^{pq} is defined, such that x_{ik}^{pq} is 1 if haplotype h_i^p of genotype g_i and haplotype h_k^q of genotype g_k are different. Two incompatible genotypes are guaranteed not to be explained by the same haplotype, and therefore, for the four possible combinations of p and q, $x_{ik}^{pq} = 1$. Moreover, two genotypes g_i and g_k are related only to respect to sites for which either g_i or g_k is heterozygous at that site. Therefore, the conditions on the x_{ik}^{pq} variables are all of the following form, for all $1 \leq j \leq m$,

$$\neg(R \Leftrightarrow S) \Rightarrow x_{ik}^{pq}, \tag{7.15}$$

where the propositions R and S depend on the values of the sites g_{ij} and g_{kj}, and also on the haplotype being considered, i.e., either h^a or h^b. Observe that $1 \leq k < i \leq n$, $1 \leq j \leq m$, and $p, q \in \{a, b\}$. Accordingly, the R and S propositions are defined as follows:

- If $g_{ij} \neq 2 \wedge g_{kj} = 2$, then $R = (g_{ij} \Leftrightarrow (q \Leftrightarrow a))$ and $S = t_{kj}$.

- If $g_{kj} \neq 2 \wedge g_{ij} = 2$, then $R = (g_{kj} \Leftrightarrow (p \Leftrightarrow a))$ and $S = t_{ij}$.
- If $g_{ij} = 2 \wedge g_{kj} = 2$, then $R = (p \Leftrightarrow q)$ and $S = (t_{ij} \Leftrightarrow t_{kj})$.

Moreover, in order to count the number of distinct haplotypes used, Boolean variables u_i^p are defined such that u_i^p is assigned value 1 if haplotype h_i^p explaining genotype g_i is different from every haplotype which explain genotype g_k with $k<i$. The conditions on variables u_i^p are based on the conditions for the x_i variables,

$$\bigwedge_{1\leq k<i} \left(x_{ik}^{pa} \wedge x_{ik}^{pb}\right) \Rightarrow u_i^p. \tag{7.16}$$

Finally, the cost function is given by

$$\min \sum_{i=1}^{n} \left(u_i^a + u_i^b\right). \tag{7.17}$$

Theorem 7.2. *(Space Complexity) Let n be the number of genotypes in $\mathcal{G}$, m the number of positions of each genotype and θ the number of heterozygous positions of the instance. Then, the number of variables of the PBO model is $\mathcal{O}(n^2 + \theta)$, which corresponds to $\mathcal{O}(n^2 + nm)$. In addition, the number of constraints of the PBO model is $\mathcal{O}(n^2 m)$.*

The RPoly model has been further improved, following the ideas used by SHIPs. In particular, the utilization of lower bounds as a method for pruning the search space shows to be very effective. The integration of a tight lower bound (Sect. 7.5.2) allows fixing the values of some variables u, and the clauses used for constraining the value of u need not be generated. This allows the PBO solver to focus on the remaining u variables, and the size of the generated PBO problem instances becomes significantly smaller. For the more complex problem instances, the integration of lower bound information reduces the size of the generated PBO instances by a factor between 2 and 3, on average.

Finally, an additional improvement is the inclusion of cardinality constraints on the x variables. Adding new constraints to a model is expected to prune the search and therefore contributes to the efficiency of the solver.

7.4.4.1 Missing Data

Most often real genotype data contain a significant percentage of unknown data. Even with modern automated DNA analysis techniques, generating data with missing alleles is not an uncommon situation [25].

One useful feature of the RPoly tool is to be able to deal with unspecified genotype sites. Genotyping procedures often leave a percentage of missing genotype positions, and so haplotype inference tools need to be able to deal with missing sites. RPoly can handle SNPs with unspecified values, inferring the values for the missing sites and still guaranteeing a parsimonious solution.

The formulation is as follows. Two Boolean variables are associated with each missing site to represent the four possible values for the haplotypes: two homozygous values (one for each allele) and two heterozygous values (one for each haplotype phase). The constraints for unspecified genotype sites are similar to the constraints for heterozygous genotype sites.

Missing SNPs are represented by "?". Hence, the alphabet of the genotypes is extended to $\{0, 1, 2, ?\}$. In practice, two variables, t_{ij}^a and t_{ij}^b, are associated with each missing site $g_{ij} =?$. Then, $t_{ij}^p = 0$ indicates that $h_{ij}^p = 0$, whereas $t_{ij}^p = 1$ indicates that $h_{ij}^p = 1$, with $p \in \{a, b\}$.

The conditions on the $x_{i\,k}^{p\,q}$ variables, which correspond to (7.15), are updated to

$$\neg(R \Leftrightarrow S) \Rightarrow x_{ik}^{pq}, \tag{7.18}$$

where the propositions R and S depend on the values of the sites g_{ij} and g_{kj}, and also on which of the haplotypes is being considered, i.e., either a or b. Note that $1 \leq k < i \leq n$, $1 \leq j \leq m$, and $p, q \in \{a, b\}$. Accordingly, the R and S propositions are defined as follows:

- If $g_{ij} \neq 2 \wedge g_{kj} = 2$, then $R = (g_{ij} \Leftrightarrow (q \Leftrightarrow a))$ and $S = t_{kj}$.
- If $g_{kj} \neq 2 \wedge g_{ij} = 2$, then $R = (g_{kj} \Leftrightarrow (p \Leftrightarrow a))$ and $S = t_{ij}$.
- If $g_{ij} = 2 \wedge g_{kj} = 2$, then $R = (p \Leftrightarrow q)$ and $S = (t_{ij} \Leftrightarrow t_{kj})$.
- If $g_{ij} = ? \wedge g_{kj} \notin \{2, ?\}$, then $R = t_{ij}^p$ and $S = g_{kj}$.
- If $g_{kj} = ? \wedge g_{ij} \notin \{2, ?\}$, then $R = t_{kj}^q$ and $S = g_{ij}$.
- If $g_{ij} = ? \wedge g_{kj} = 2$, then $R = (q \Leftrightarrow a)$ and $S = (t_{ij}^p \Leftrightarrow t_{kj})$.
- If $g_{kj} = ? \wedge g_{ij} = 2$, then $R = (p \Leftrightarrow a)$ and $S = (t_{kj}^q \Leftrightarrow t_{ij})$.
- If $g_{ij} = ? \wedge g_{kj} =?$, then $R = t_{ij}^p$ and $S = t_{kj}^q$.

7.5 Standard Techniques

This section describes some techniques which can be applied before applying any HIPP solver. These techniques are inexpensive and empirical evidence shows that they can significantly improve the performance of the solvers.

7.5.1 Structural Simplifications

Structural simplifications to the set of genotypes can be performed for all HIPP solvers as a preprocessing technique. These simplifications use the structural properties of genotypes with the purpose of reducing the search space [3, 31].

We start by observing that two equal genotypes can be explained identically, maintaining a HIPP solution, and, consequently, one of them can be discarded. The solution of the discarded genotype is assumed to be the solution of the remaining genotype. In addition, duplicate sites can be discarded. If all genotypes have the same values on a pair of sites, then one of the sites can be removed. Moreover, complemented sites can also be discarded. Two sites are complemented if whenever one site has value 1, the other site has value 0 and vice versa.

By keeping information of the genotypes and sites discarded, it is straightforward to construct a solution for the original set of genotypes once discovered a solution to the simplified set of genotypes.

Example 7.4. (Structural Simplifications) Consider the following set of four genotypes each with five sites: $\mathcal{G} = \{02122,\ 12021,\ 02122,\ 20201\}$. Note that the first genotype is equal to the third genotype. Hence, the third genotype can be discarded. In addition, note that the second and the fourth sites are equal. Also the first and the third sites are complemented. Hence, the third and the fourth sites can be removed. Consequently, the simplified set of genotypes is a set with three genotypes and three sites: *simplified*$(\mathcal{G}) = \{022,\ 121,\ 201\}$.

7.5.2 Lower Bounds

The computation of tight bounds is a key issue in several solvers, e.g., SHIPs and HAPAR. This section describes two algorithms used to compute lower bounds which have been proposed with the SHIPs algorithm [29, 31].

The first algorithm relies on the incompatibility between genotypes [29]. Recall that two genotypes are *incompatible* if there exists a site for which the value of one genotype is 0 and the other genotype is 1. Otherwise, the genotypes are *compatible*. Clearly, incompatible genotypes cannot be explained by common haplotypes. The first lower bound procedure consists in finding a clique of incompatible genotypes in a graph where each vertex is a genotype, and there is an edge between two genotypes if they are incompatible. If the clique has k genotypes, then the number of haplotypes required is guaranteed to be equal or greater than $2k - \sigma$, where σ is the number of homozygous genotypes in the clique.

The clique lower bound previously described can be improved taking into account the structure of the genotypes [31]. Let G_S denote the set of selected genotypes. At the beginning, G_S corresponds to the set of genotypes in the clique of incompatible genotypes. The goal is to find heterozygous sites g_{ij} in genotypes which do not belong to G_S, such that all genotypes $g \in G_\mathrm{S}$ compatible with g_i have the same homozygous value in that position. Define the propositional function $\kappa(i, k)$ to be true if g_i and g_k are compatible. (Clearly, κ is a symmetric relation.) Thus, the goal is to find a genotype $g_i \notin G_\mathrm{S}$ such that there exists a position j where $g_{ij} = 2$ and a value val $\in \{0, 1\}$ such that, for all $g_k \in G_\mathrm{S}$, if $\kappa(i, k)$ then $g_{kj} =$ val, i.e.,

$$\exists_{1 \leq j \leq m}(g_{ij} = 2 \wedge \exists_{\mathrm{val} \in \{0,1\}} \forall_{g_k \in G_\mathrm{S}} (\kappa(i, k) \Rightarrow g_{kj} = \mathrm{val})). \tag{7.19}$$

Then $G_\mathrm{S} = G_\mathrm{S} \cup \{g_i\}$ and the lower bound can be increased by one. The process is iterated until there are no more genotypes $g_i \notin G_\mathrm{S}$ which satisfy (7.19).

The interest in integrating a lower bound in SHIPs is twofold. First, the number of iterations of the algorithm is reduced. Second, and more important, the size of the

SAT formula can be significantly reduced based on the information achieved with the computation of the lower bound.

Example 7.5. (Lower Bounds) Consider the set of genotypes $\mathcal{G} = \{g_1, g_2, g_3, g_4, g_5\}$, where $g_1 = 1010$, $g_2 = 0212$, $g_3 = 1210$, $g_4 = 2120$, $g_5 = 2222$. A clique of incompatible genotypes is $\{g_1,\ g_2\}$, where g_2 contributes with 2 to the lower bound and g_1 contributes with 1 because g_1 is homozygous. Therefore, these genotypes are selected and the value of the clique lower bound is 3. The second lower bound increases the value of the lower bound by 2. Genotypes g_3 and g_4 are selected because they are heterozygous at positions where every genotype already selected and compatible with them have the same homozygous value. Consequently, at least five different haplotypes $\{h_1, h_2, h_3, h_4, h_5\}$ are required to explain $\mathcal{G}$. Genotype g_1 can be associated with $\{h_1,\ h_1\}$ and g_2 can be associated with $\{h_2,\ h_3\}$. Genotypes g_3 and g_4 can be associated, respectively, with h_4 and h_5. Only g_5 does not contribute to increasing the lower bound.

7.5.3 Upper Bounds

The computation of upper bounds is also an important issue in some solvers [1, 44]. In general, every heuristic algorithm to the HIPP problem can be used to produce an upper bound, for example, the delayed haplotype selection (DS) algorithm [34] and the CollHaps heuristic approach [43]. DS was suggested for using binary search in SHIPs and CollHaps is used by some exact models [1]. Both DS and Collhaps are based on the Clark's method.

Clark's method [6] is a well-known algorithm to solve the haplotype inference problem. This method starts by identifying genotypes with zero or one heterozygous sites, which have only one possible explanation. Then, the method attempts to explain the remaining genotypes with at least one of the haplotypes already identified (Clark's rule). This may eventually require the inference of new haplotypes which will be added to the set of haplotypes. The key point to note is that there are many ways to extend the set of haplotypes, since for genotypes with more than one heterozygous site there are a few possible explanations.

Clearly, the Clark's method may be used to compute an upper bound to the HIPP problem. However, this method is often too greedy. DS addresses the main drawback of Clark's method, avoiding the excessive greediness. Collhaps is based on a generalization of the Clark's rule, called the collapse rule. These two heuristic approaches provide very accurate approximation to the pure parsimony solution.

7.6 Experimental Evaluation

This section presents a summary of the experimental results, obtained in a set of 1183 instances, including both synthetic and real data. Synthetic data include instances generated using the Hudson's program, *ms* [23], and harder instances described in [39] which were generated to evaluate phasing algorithms. Real data were obtained from the HapMap project and biological instances generated from publicly available data (e.g., [9, 10, 26, 38]). All problem instances were simplified in a preprocessing step, as described in Sect. 7.5.1. The maximum number of SNPs on the simplified instances is 188 and the maximum number of genotypes is 94. For a more detailed description of problem instances and results, see other literature [17, 31, 32].

The extensive comparison includes nine HIPP solvers: RPoly [17] version 1.2, SHIPs [31] version 2, HaploPPH [5], Haplo-ASP [12], RTIP [19], SATlotyper [36] version 0.1.1 b, HAPAR [44], PolyIP [2] and HybridIP [3]. Results were obtained on an Intel Xeon 5160 server (3.0GHz, 4GB RAM) running Red Hat Enterprise Linux WS 4. The CPU time is limited to 1,000 s.

Table 7.1 resumes the performance of exact HIPP solvers, stating the percentage and the mean time required for solving the solved instances within 1,000 s. RPoly is the HIPP tool capable of solving the largest number of instances. Clearly, the SAT-based models are the best performing solvers. RPoly solves 1,165 instances and SHIPs solves 1,116 of 1,183 instances. Taking into account the amount of time that each solver requires, on average, to solve the non-aborted instances, we can conclude that RPoly is also the fastest solver. The exponential ILP model, RTIP, is significantly efficient for easy instances. Nonetheless, RTIP is not able to solve more than 30% of the instances due to memory exhaustion. The performance of PolyIP and HybridIP is similar, both solving around 40% of the instances.

SATlotyper is a more general HIPP solver, being able to solve haplotype inference problems on polyploids and polyallelic species. Therefore, SATlotyper was not created to compete with exact HIPP solvers and is less optimized than SHIPs. In particular, SATlotyper does not include the computation of the lower bound, which is a crucial point to the good performance of SHIPs.

The fact that the best performing solvers are RPoly and SHIPs suggests that the SAT-based techniques are the most adequate for solving the HIPP problem.

Table 7.1 Summary of the performance of exact HIPP solvers

Model	% solved instances	Average time (s)
RPoly	98.5%	3.53
SHIPs	94.3%	7.86
HaploPPH	79.1%	42.34
Haplo-ASP	73.5%	30.18
RTIP	68.0%	6.78
SATlotyper	66.9%	24.41
HAPAR	48.9%	39.90
PolyIP	40.2%	73.44
HybridIP	39.6%	72.94

7.7 Conclusions

Haplotype inference is an important biological problem with relevant implications in medical studies. Haplotype inference aims at deriving the genetic constitution of a single chromosome (haplotype), given the conflated data of a pair of homologous chromosomes (genotype). The HIPP problem is an optimization approach which seeks the solution that minimizes the total number of used haplotypes. This problem is APX-hard. A significant number of models have been proposed to solve the problem.

Models based on SAT represent notable approaches to HIPP. Starting from a lower bound on the number of required haplotypes, SHIPs model the haplotype inference problem into a SAT formula and use a SAT solver to obtain a solution. The algorithm proceeds iteratively until a satisfiable assignment is obtained, which guarantees the minimum number of haplotypes. This iterative algorithm is necessary because SAT solvers do not handle optimization functions. The extension of SAT to handle cost functions is known as PBO. The PBO model to HIPP is called RPoly. The RPoly model integrates important features of previous models into a new approach.

Both SAT-based methods, SHIPs and RPoly, are considerably more efficient than the remaining HIPP algorithms and constitute the state of the art in the field.

Acknowledgments This research was funded by Fundação para a Ciência e Tecnologia under research project SHIPs (PTDC/EIA/64164/2006) and PhD grant (SFRH/BD/28599/2006), and by Microsoft under contract 2007-017 of the Microsoft Research PhD Scholarship Programme.

References

1. P. Bertolazzi, A. Godi, M. Labbé, and L. Tininini. Solving haplotyping inference parsimony problem using a new basic polynomial formulation. *Computers & Mathematics with Applications*, 55(5):900–911, 2008
2. D. Brown and I. Harrower. A new integer programming formulation for the pure parsimony problem in haplotype analysis. In *Workshop on Algorithms in Bioinformatics (WABI'04)*, volume 3240 of *LNCS*, pages 254–265, 2004
3. D. Brown and I. Harrower. Integer programming approaches to haplotype inference by pure parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(2):141–154, 2006
4. S. Browning and B. Browning. Rapid and accurate haplotype phasing and missing data inference for whole genome association studies using localized haplotype clustering. *American Journal of Human Genetics*, 81(5):1084–1097, 2007
5. D. Catanzaro, A. Godi, and M. Labbé. A class representative model for pure parsimony haplotyping. *INFORMS Journal on Computing*, 22(2):195–209, 2009
6. A. G. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–122, 1990
7. S. A. Cook. The complexity of theorem-proving procedures. In *ACM Symposium on Theory of Computing (STOC'71)*, pages 151–158, 1971
8. D. C. Crawford and D. A. Nickerson. Definition and clinical importance of haplotypes. *Annual Review of Medicine*, 56:303–320, 2005

9. M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and E. S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29:229–232, 2001
10. C. M. Drysdale, D. W. McGraw, C. B. Stack, J. C. Stephens, R. S. Judson, K. Nandabalan, K. Arnold, G. Ruano, and S. B. Liggett. Complex promoter and coding region β_2-adrenergic receptor haplotypes alter receptor expression and predict in vivo responsiveness. In *National Academy of Sciences*, volume 97, pages 10483–10488, 2000
11. N. Eén and N. Sörensson. An extensible SAT-solver. In *International Conference on Theory and Applications of Satisfiability Testing (SAT'03)*, pages 502–518, 2003
12. E. Erdem and F. Ture. Efficient haplotype inference with answer set programming. In *National Conference on Artificial Intelligence (AAAI'08)*, pages 434–441, 2008
13. L. Gaspero and A. Roli. Stochastic local search for large-scale instances of the haplotype inference problem by pure parsimony. *Journal of Algorithms*, 63(1–3):55–69, 2008
14. E. Giunchiglia, Y. Lierler, and M. Maratea. Answer set programming based on propositional satisfiability. *Journal of Automated Reasoning*, 36(4):345–377, 2006
15. D. B. Goldstein, S. K. Tate, and S. M. Sisodiya. Pharmacogenetics goes genomic. *Nature Reviews Genetics*, 4(12):937–947, 2003
16. A. Graça, J. Marques-Silva, I. Lynce, and A. Oliveira. Efficient haplotype inference with pseudo-Boolean optimization. In *Algebraic Biology (AB'07)*, volume 4545 of *LNCS*, pages 125–139, 2007
17. A. Graça, J. Marques-Silva, I. Lynce, and A. Oliveira. Haplotype inference with pseudo-Boolean optimization. *Annals of Operations Research*, doi:10.1007/s10479-009-0675-4, 2010 (in Press) http://www.springerlink.com/content/f8p2583387721p5t/
18. D. Gusfield. Haplotyping as perfect phylogeny: conceptual framework and efficient solutions. In *International Conference on Research in Computational Molecular Biology (RECOMB'02)*, pages 166–175, 2002
19. D. Gusfield. Haplotype inference by pure parsimony. In *Annual Symposium on Combinatorial Pattern Matching (CPM'03)*, pages 144–155, 2003
20. B.V. Halldórsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail. A survey of computational methods for determining haplotypes. In *DIMACS/RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotype Inference*, volume 2983 of *LNCS*, pages 26–47, 2004
21. Y-T. Huang, K-M. Chao, and T. Chen. An approximation algorithm for haplotype inference by maximum parsimony. *Journal of Computational Biology*, 12(10):1261–1274, 2005
22. R. Hudson. Gene genealogies and the coalescent process. *Oxford Survey of Evolutionary Biology*, 7:1–44, 1990
23. R. R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002
24. G. Johnson, L. Esposito, B. Barratt, A. Smith, J. Heward, G. Genova, H. Ueda, H. Cordell, I. Eaves, F. Dudbridge, R. Twells, F. Payne, W. Hughes, S. Nutland, H. Stevens, P. Carr, E. Tuomilehto-Wolf, J. Tuomilehto, S. Gough, D. Clayton, and J. Todd. Haplotype tagging for the identification of common disease genes. *Nature*, 29:233–237, 2001
25. E. Kelly, F. Sievers, and R. McManus. Haplotype frequency estimation error analysis in the presence of missing genotype data. *BMC Bioinformatics*, 5:188, 2004
26. D. L. Kroetz, C. Pauli-Magnus, L. M. Hodges, C. C. Huang, M. Kawamoto, S. J. Johns, D. Stryke, T. E. Ferrin, J. DeYoung, T. Taylor, E. J. Carlson, I. Herskowitz, K. M. Giacomini, and A. G. Clark. Sequence diversity and haplotype structure in the human ABCD1 (MDR1, multidrug resistance transporter). *Pharmacogenetics*, 13:481–494, 2003
27. G. Lancia and P. Serafini. A set-covering approach with column generation for parsimony haplotyping. *INFORMS Journal on Computing*, 21(1):151–166, 2009
28. G. Lancia, C. M. Pinotti, and R. Rizzi. Haplotyping populations by pure parsimony: complexity of exact and approximation algorithms. *INFORMS Journal on Computing*, 16(4):348–359, 2004
29. I. Lynce and J. Marques-Silva. Efficient haplotype inference with Boolean satisfiability. In *National Conference on Artificial Intelligence (AAAI'06)*, pages 104–109, 2006

30. I. Lynce and J. Marques-Silva. SAT in bioinformatics: Making the case with haplotype inference. In *International Conference on Theory and Applications of Satisfiability Testing (SAT'06)*, volume 4121 of *LNCS*, pages 136–141, 2006
31. I. Lynce and J. Marques-Silva. Haplotype inference with Boolean satisfiability. *International Journal on Artificial Intelligence Tools*, 17(2):355–387, 2008
32. I. Lynce, A. Graça, J. Marques-Silva, and A. Oliveira. Haplotype inference with Boolean constraint solving: an overview. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI'08)*, volume I, pages 92–100, 2008
33. J. Marques-Silva. Practical applications of Boolean satisfiability. In *Workshop on Discrete Event Systems (WODES'08)*, 2008
34. J. Marques-Silva, I. Lynce, A. Graça, and A. Oliveira. Efficient and tight upper bounds for haplotype inference by pure parsimony using delayed haplotype selection. In *13th Portuguese Conference on Artificial Intelligence (EPIA'07)*, volume 4874 of *LNAI*, pages 621–632. Springer, 2007
35. J. McCluskey and C. A. Peh. The human leucocyte antigens and clinical medicine: an overview. *Reviews in Immunogenetics*, 1(1):3–20, 1999
36. J. Neigenfind, G. Gyetvai, R. Basekow, S. Diehl, U. Achenbach, C. Gebhardt, J. Selbig, and B. Kersten. Haplotype inference from unphased SNP data in heterozygous polyploids based on SAT. *BMC Genomics*, 9:356, 2008
37. X. Pan. Haplotype inference by pure parsimony with constraint programming. Master's thesis, Faculty of Science and Technology, Uppsala Universitet, Sweden, 2009
38. M. J. Rieder, S. T. Taylor, A. G. Clark, and D. A. Nickerson. Sequence variation in the human angiotensin converting enzyme. *Nature Genetics*, 22:481–494, 2001
39. S.F. Schaffner, C. Foo, S. Gabriel, D. Reich, M.J. Daly, and D. Altshuler. Calibrating a coalescent simulation of human genome sequence variation. *Genome Research*, 15:1576–1583, 2005
40. S. T. Sherry, M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29:308–311, 2001
41. M. Stephens, N. Smith, and P. Donelly. A new statistical method for haplotype reconstruction. *American Journal of Human Genetics*, 68:978–989, 2001
42. The International HapMap Consortium. A second generation human haplotype map over 3.1 million snps. *Nature*, 449:851–861, 2007
43. L. Tininini, P. Bertolazzi, A. Godi, and G. Lancia. CollHaps: A heuristic approach to haplotype inference by parsimony. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 99(1), 2008
44. L. Wang and Y. Xu. Haplotype inference by maximum parsimony. *Bioinformatics*, 19(14):1773–1780, 2003
45. R.-S. Wang, X.-S. Zhang, and L. Sheng. Haplotype inference by pure parsimony via genetic algorithm. In *International Symposium on Operations Research and Its Applications (ISORA'05)*, pages 308–318, 2005

Chapter 8
Estimating Phylogenies from Molecular Data

Daniele Catanzaro

Abstract Phylogenetic estimation from aligned DNA, RNA or amino acid sequences has attracted more and more attention in recent years due to its importance in analysis of many fine-scale genetic data. Nowadays, its application fields range from medical research to drug discovery, to epidemiology, to systematics and population dynamics. Estimating phylogenies involves solving an optimization problem, called the phylogenetic estimation problem (PEP), whose versions depend on the criterion used to select a phylogeny among plausible alternatives. This chapter offers an overview of PEP and discuss the most important versions that occur in the literature.

8.1 Introduction

Molecular phylogenetics studies the hierarchical evolutionary relationships among species, or *taxa*, by means of molecular data such as DNA, RNA, amino acid or codon sequences. These relationships are usually described through a weighted tree, called a *phylogeny* (Fig. 8.1), whose *leaves* represent the observed taxa, *internal vertices* represent the intermediate ancestors, *edges* represent the estimated evolutionary relationships and *edge weights* represent measures of the similarity between pairs of taxa.

Phylogenies provide a fundamental information in analysis of many fine-scale genetic data; for this reason, the use of molecular phylogenetics has become more and more frequent (and sometimes indispensable) in several research fields such as systematics, medical research, drug discovery, epidemiology and population dynamics [56]. For example, the use of molecular phylogenetics was of considerable assistance to predict the evolution of human influenza A [8], to understand the

D. Catanzaro (✉)
Service Graphes and Mathematical Optimization, Computer Science Department, Université Libre de Bruxelles (U.L.B.), Boulevard du Triomphe, CP 210/01, B-1050 Brussels, Belgium
e-mail: dacatanz@ulb.ac.be

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_8,

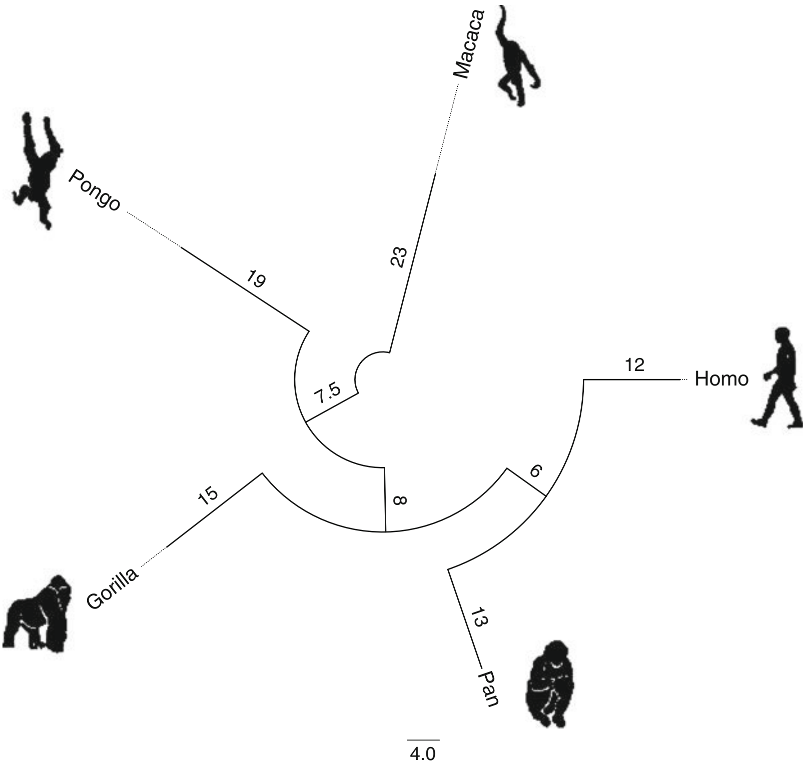


Fig. 8.1 An example of a phylogeny of primates

relationships between the virulence and the genetic evolution of HIV [55, 66], to identify emerging viruses as SARS [51], to recreate and investigate ancestral proteins [17], to design neuropeptides causing smooth muscle contraction [2] and to relate geographic patterns to macroevolutionary processes [36].

Since no one could observe evolution over thousands or millions of years, a part from known phylogenies ([57]), there is no general way to validate empirically a candidate phylogeny for a set of molecular sequences extracted from taxa. For this reason, the literature proposes a number of criteria for selecting one phylogeny from among plausible alternatives. Each criterion adopts its own set of evolutionary hypotheses, whose ability to describe evolution of taxa determines the gap between the *real* and the *true phylogeny*, i.e., the gap between the real evolutionary process of taxa and the phylogeny that one would obtain under the same set of hypotheses if all molecular data from taxa were available [9].

The criteria of phylogenetic estimation can usually be quantified and expressed in terms of objective functions, giving rise to families of optimization problems whose general paradigm can be stated as follows:

Problem 8.1. The phylogenetic estimation problem (PEP)

$$\begin{aligned} \text{optimize} \quad & f(T) \\ \text{s.t.} \quad & g(\Gamma, T) = 0 \\ & T \in \mathcal{T}, \end{aligned}$$

where Γ is the set of molecular sequences from n taxa, T a phylogeny of Γ, $\mathcal{T}$ the set of $(2n-5)!! = 1\times 3\times 5\times 7\cdots\times 2n-5$ phylogenies of Γ, $f : \mathcal{T} \to \mathbb{R}$ a function modeling the selected criterion of phylogenetic estimation, and $g : \Gamma \times \mathcal{T} \to \mathbb{R}$ a function correlating the set Γ to a phylogeny T.

A specific optimization problem, or *phylogenetic estimation paradigm*, is completely characterized by defining the functions f and g. The phylogeny T^* that optimizes f and satisfies g is referred to as *optimal*, and if T^* approaches the true phylogeny as the amount of molecular data from taxa increases, the corresponding criterion is said to be *statistically consistent* [32]. The statistical consistency is a desirable property in molecular phylogenetics because it measures the ability of a criterion to recover the true (and hopefully the real) phylogeny of the given molecular data. Later in this chapter, we will show that the consistency property changes from criterion to criterion and in some cases may be even absent.

Here, we provide a review of the main estimation criteria that occur in the literature on molecular phylogenetics. Particular emphasis is given to the comparative description of the hypotheses at the core of each criterion and to the optimization aspects related to the phylogenetic estimation paradigms. In Sect. 8.2, we discuss the problem of measuring the similarity among molecular sequences. In Sect. 8.3, we discuss the fundamental least-squares paradigm and formalize the concept of phylogeny. In Sect. 8.4, we present the minimum evolution paradigm by evidencing the recent perspectives and computational advances. Finally, in Sect. 8.5 we present the likelihood and the bayesian paradigms by exposing briefly their benefits and drawbacks.

8.2 Measuring Molecular Similarity

The degree of similarity between pairwise molecular sequences reflects the amount of mutation events that occurred since they split from their common ancestor. Quantifying such similarity constitutes the first step in the phylogenetic estimation process [11]. The task involves the investigation and the modeling of the *mutation process* over time, i.e., the process by which errors occur in molecular data and are inherited between generations.

Different types of mutation may occur in the genome structure, most of which are point mutations, i.e., changes that involve the replacement, or *substitution*, of one nucleotide for another in the DNA sequence. Point mutations can be classified in two categories: the transitions and the transversions. The transitions occur when a purine nucleotide (adenine or guanine) is substituted for another purine, or when a pyrimidine (cytosine or thymine) is substituted for another pyrimidine. The transversions occur when a pyrimidine is substituted for a purine, or vice versa.

A second class of point mutations are those that lead to *insertions* and *deletions* of nucleotides in the genome. This phenomenon mainly occurs in non-coding regions of DNA, but may interest also coding regions of the genome and be the cause of deleterious effects [57].

Finally, a third class of mutations are those that involve entire chromosome regions of the genome. Specifically, we may have: (1) a *duplication*, when a chromosome region is duplicated; (2) a *translocation*, when a chromosome region is transferred into another chromosome; (3) an *inversion*, when a chromosome region is broken off, turned upside down and reconnected; (4) a *deletion*, when a chromosome region is missing or deleted; (5) and a *loss of heterozygosity*, e.g., when two instances of the same chromosome break and then reconnect but to the different end pieces [57].

Modeling the second and the third classes of mutations is generally non-trivial and requires advanced mathematical background. We refer the interested reader to Felsenstein [29] for an introduction and to Park and Deem [58] for recent advances in the modeling of such classes. Here we shall focus on the first class of mutations and present a fundamental model of molecular evolution which is at the core of the most currently used criteria of phylogenetic estimation. Unless otherwise stated, throughout the chapter we will always assume that the molecular sequences under study have been previously subjected to an *alignment process*, i.e., a process through which the evolutionary relationships between nucleotides of molecular data are evidenced (see [60] for details).

8.2.1 The Time Homogeneous Markov Model of Molecular Evolution

Let S be a DNA sequence, i.e., a string of fixed length over an alphabet $\Upsilon = \{A, C, G, T\}$, where "A" codes for adenine, "C" for cytosine, "G" for guanine, and "T" for thymine. Let $r_{ij} \geq 0$, $i \neq j$, be the constant rate of substitution from nucleotide i to nucleotide j. Assume that each character (site) of S evolves independently over time and that, instant per instant, the *Markov conservative hypothesis* [39] holds, i.e.,

$$r_{ii} = -\sum_{j \in \Upsilon,\ j \neq i} r_{ij} \quad \forall\, i \in \Upsilon. \tag{8.1}$$

Let $p_{ij}(t)$ be the probability that nucleotide i undergoes to a substitution to nucleotide j at finite time t. Then, if the superposition principle holds, at $t + \mathrm{d}t$ such probability can be written as:

$$p_{ij}(t + \mathrm{d}t) = \sum_{k \in \Upsilon} p_{ik}(t) p_{kj}(\mathrm{d}t) \quad \forall\, i, j \in \Upsilon. \tag{8.2}$$

By subtracting $p_{ij}(t)$ in both sides of (8.2) and dividing for dt we obtain:

$$\begin{aligned}\frac{p_{ij}(t + \mathrm{d}t) - p_{ij}(t)}{\mathrm{d}t} = {} & \frac{\sum_{k \in \Upsilon,\ k \neq j} p_{ik}(t) p_{kj}(\mathrm{d}t)}{\mathrm{d}t} \\ & + p_{ij}(t) \frac{p_{jj}(\mathrm{d}t) - 1}{\mathrm{d}t} \quad \forall\, i, j \in \Upsilon,\end{aligned}$$

i.e.,

$$\begin{aligned}\frac{p_{ij}(t + \mathrm{d}t) - p_{ij}(t)}{\mathrm{d}t} = {} & \frac{\sum_{k \in \Upsilon,\ k \neq j} p_{ik}(t) p_{kj}(\mathrm{d}t)}{\mathrm{d}t} \\ & + p_{ij}(t) \frac{1 - \sum_{k \in \Upsilon,\ k \neq j} p_{kj}(\mathrm{d}t) - 1}{\mathrm{d}t} \quad \forall\, i, j \in \Upsilon.\end{aligned}$$

Hence, we have

$$\dot{p}_{ij}(t) = \sum_{k \in \Upsilon,\ k \neq j} p_{ik}(t) r_{kj} + p_{ij}(t) r_{jj} \quad \forall\, i, j \in \Upsilon. \tag{8.3}$$

When expressing (8.3) in matrix form, the Chapman–Kolmogorov master equation arises

$$\dot{\mathbf{P}}(t) = \mathbf{P}(t)\mathbf{R} = \mathbf{R}\mathbf{P}(t),$$

whose integral

$$\mathbf{P}(t) = \mathbf{e}^{\mathbf{R}t} = \sum_{n=0}^{\infty} \frac{\mathbf{R}^n t^n}{n!} \tag{8.4}$$

is known as the time homogeneous Markov (THM) model of DNA sequence evolution [48, 63]. The THM model is a generalization of the Markov models described in Jukes and Cantor [44], Kimura [46], Hasegawa et al. [37], Tamura and Nei [78], and can be easily adapted to RNA, amino acid and codon sequences as shown in Felsenstein [29] and Schadt and Lange [71, 72]. In the next section, we shall investigate the dynamics of the THM model in order to derive a commonly used formula to quantify the similarity between molecular data.

8.2.2 Estimating Evolutionary Distances from Molecular Data

Two molecular sequences S_1 and S_2, evolving at time t_0 from a common ancestor, could be characterized at time t by different amounts of substitution events, some of which not directly observable. Hence, if we would sample the sequences at time t and measure their similarity, or *evolutionary distance*, in terms of number of observed differences, we could underestimate the overall substitution events that occurred since S_1 and S_2 split from their common ancestor. A number of authors suggested that the use of the time homogeneous Markov models could overcome the underestimation problem in all those cases in which the hypotheses at the core of the model would properly describe the real evolutionary process of the analyzed sequences [29]. Moreover, in order to compare the evolutionary distances of different pairs of molecular sequences, the authors also proposed to express the evolutionary distances in terms of expected number of substitution events per site rather than the time necessary to transform a sequence into another [29]. In this section, we will present the most general formula currently known in the literature to compute the evolutionary distance from pairwise molecular sequences. To this aim, we shall investigate now the dynamics of the THM model.

As shown in Zadeh and Desoer [84], (8.4) can also be expressed in closed formula as:

$$\mathbf{P}(t) = \mathbf{e}^{\mathbf{R}t} = \Omega \mathbf{e}^{\Lambda t} \Omega^{-1}, \tag{8.5}$$

where Ω is the eigenvector matrix of $\mathbf{R}$, and Λ is the diagonal matrix of the eigenvalues of $\mathbf{R}$. This fact suggests that the spectrum of $\mathbf{P}(t)$ is the exponential spectrum of $\mathbf{R}$, i.e., the dynamics of $\mathbf{P}(t)$ is univocally determined from the knowledge of the spectrum of $\mathbf{R}$ [84].

It is worth noting that the Markov conservative hypothesis implies that the determinant of matrix $\mathbf{R}$ is equal to zero, i.e., at least one of its eigenvalues is identically zero. Moreover, since any k-leading principal sub-matrix of $\mathbf{R}$, $k < 4$, has negative determinant, for one of the Sylvester corollaries (see [6, p. 409]) all the remaining eigenvalues are negative. Thus, as the spectrum of $\mathbf{P}(t)$ is the exponential spectrum of $\mathbf{R}$, matrix $\mathbf{P}(t)$ has at least one eigenvalue equal to 1, called *the maximal Lyapunov exponent*, and three eigenvalues lying in the interval [0, 1]. The maximal Lyapunov exponent prevents the presence of chaotic attractors and guarantees that, as t goes to infinity, the generic entry $p_{ij}(t)$ is non-zero and independent on the starting state $i \in \Upsilon$. In other words, the maximal Lyapunov exponent guarantees the existence of four positive values π_A, π_C, π_G, and π_T, called *equilibrium frequencies*, such that

$$\lim_{t \to \infty} p_{ij}(t) = \pi_j \qquad \forall\, i, j \in \Upsilon.$$

The values π_j constitute a *stationary distribution* and turn out to be useful to measure the evolutionary distance between S_1 and S_2. In fact, denote $\mathbf{O}(t)$ as a matrix whose generic entry $o_{ij}(t)$, $i, j \in \Upsilon$, represents the probability that at a given site and time t, S_1 is characterized by nucleotide i and S_2 by nucleotide j. Assume that $\mathbf{O}(0) = \Pi$, where Π denotes a diagonal matrix whose jth diagonal entry is π_j. Then it holds that:

$$o_{ij}(t) = \sum_{k \in \Upsilon} p'_{ik}(t)\pi_k p_{kj}(t) \qquad \forall\, i, j \in \Upsilon,\ t \geq 0,$$

or equivalently

$$\mathbf{O}(t) = \mathbf{P}'(t)\Pi\mathbf{P}(t) \qquad t \geq 0, \tag{8.6}$$

where $\mathbf{P}'(t)$ denotes the transpose of $\mathbf{P}(t)$. Premultiplying for Π^{-1} both sides of (8.6) we have

$$\Pi^{-1}\mathbf{O}(t) = \Pi^{-1}\mathbf{P}'(t)\Pi\mathbf{P}(t) = \Pi^{-1}\mathbf{e}^{\mathbf{R}'t}\Pi\mathbf{e}^{\mathbf{R}t}.$$

Since for any matrix function it holds that $f(\mathbf{ABA}^{-1}) = \mathbf{A}f(\mathbf{B})\mathbf{A}^{-1}$, we have

$$\Pi^{-1}\mathbf{O}(t) = \mathbf{e}^{\Pi^{-1}\mathbf{R}'t\Pi}\mathbf{e}^{\mathbf{R}t}. \tag{8.7}$$

If we assume that the hypothesis of *time-reversibility* holds, i.e.:

$$\Pi\mathbf{R} = \mathbf{R}'\Pi,$$

then $\Pi^{-1}\mathbf{R}'t\Pi$ and $\mathbf{R}t$ are commutative, and (8.7) becomes:

$$\Pi^{-1}\mathbf{O}(t) = \mathbf{e}^{\Pi^{-1}\mathbf{R}'t\Pi+\mathbf{R}t}. \tag{8.8}$$

By applying the logarithmic matrix function to both members of (8.8) and premultiplying for Π, we obtain

$$\mathbf{R}'t\Pi + \Pi\mathbf{R}t = \Pi\log(\Pi^{-1}\mathbf{O}(t)).$$

As the negative trace of $2t\Pi\mathbf{R}$ represents the expected number of substitution events per site between S_1 and S_2, at time t the evolutionary distance d_{S_1,S_2} between S_1 and S_2 can be computed as:

$$d_{S_1,S_2} = -2t\ \mathrm{tr}[\Pi\mathbf{R}] = -\mathrm{tr}[\Pi\log(\Pi^{-1}\mathbf{O}(t))]. \tag{8.9}$$

Equation (8.9) is known as the general time-reversible (GTR) distance [48, 63] and is the most general formula to quantify the similarity between molecular data using a time-reversible Markov model of molecular evolution. It is worth noting that if in one hand the hypothesis of time-reversibility simplifies the formalization of the evolutionary process of a pair of molecular sequences, on the other hand its introduction gives rises to important consequences. In fact, the hypothesis of time-reversibility implies that if we would compare two molecular data whose nucleotide frequencies are in equilibrium, the probability that a nucleotide i undergoes a substitution to nucleotide j would be equal to the probability that a nucleotide j undergoes a substitution to nucleotide i. Thus, given a present-day molecular sequence and its ancestral sequence, it would be impossible to determine which sequence is the present and which is the ancestral one. Hence, the hypothesis of time-reversibility removes the temporality from the evolutionary process. We shall show in the next sections how the paradigms of phylogenetic estimation take advantage of this fact. Below, we provide an example from [79] showing a possible application of (8.9).

8.2.2.1 Estimating Evolutionary Distances from Molecular Data: A Practical Example

Consider the mitochondrial DNA sequences of human and chimpanzee showed in Horai et al. [40]. The corresponding matrices $\mathbf{O}(t)$ and Π are respectively

$$\mathbf{O}(t) = \begin{array}{cc} \begin{array}{cccc} A & C & G & T \end{array} & \\ \begin{pmatrix} 0.2889 & 0.0012 & 0.0131 & 0.0005 \\ 0.0012 & 0.2799 & 0.0001 & 0.0266 \\ 0.0131 & 0.0001 & 0.1180 & 0.0001 \\ 0.00005 & 0.0266 & 0.0001 & 0.2299 \end{pmatrix} & \begin{array}{c} A \\ C \\ G \\ T \end{array} \end{array}$$

and

$$\Pi = \begin{array}{cc} \begin{array}{cccc} A & C & G & T \end{array} & \\ \begin{pmatrix} 0.3037 & 0 & 0 & 0 \\ 0 & 0.3079 & 0 & 0 \\ 0 & 0 & 0.1313 & 0 \\ 0 & 0 & 0 & 0.2571 \end{pmatrix} & \begin{array}{c} A \\ C \\ G \\ T. \end{array} \end{array}$$

The product $\Pi^{-1}\mathbf{O}(t)$ is:

$$\Pi^{-1}\mathbf{O}(t) = \begin{array}{cc} \begin{array}{cccc} A & C & G & T \end{array} & \\ \begin{pmatrix} 0.9513 & 0.0040 & 0.0430 & 0.0017 \\ 0.0040 & 0.9092 & 0.0003 & 0.0865 \\ 0.0995 & 0.0008 & 0.8989 & 0.0008 \\ 0.0030 & 0.1036 & 0.0004 & 0.8940 \end{pmatrix} & \begin{array}{c} A \\ C \\ G \\ T \end{array} \end{array}$$

and the corresponding logarithm matrix function $\log(\Pi^{-1}\mathbf{O}(t))$ is:

$$\Pi^{-1}\mathbf{O}(t) = \begin{array}{cc} \begin{array}{cccc} A & C & G & T \end{array} & \\ \begin{pmatrix} -0.0524 & 0.0042 & 0.0466 & 0.0016 \\ 0.0042 & -0.1008 & 0.0002 & 0.0963 \\ 0.1078 & 0.0006 & -0.1091 & 0.0007 \\ 0.00019 & 0.1154 & 0.0004 & -0.1176 \end{pmatrix} & \begin{array}{c} A \\ C \\ G \\ T. \end{array} \end{array}$$

The product $\Pi \log(\Pi^{-1}\mathbf{O}(t))$ is:

$$\Pi \log(\Pi^{-1}\mathbf{O}(t)) = \begin{array}{cc} \begin{array}{cccc} A & C & G & T \end{array} & \\ \begin{pmatrix} -0.0159 & 0.0013 & 0.0142 & 0.0005 \\ 0.0013 & -0.0310 & 0.0001 & 0.0297 \\ 0.0142 & 0.0001 & -0.0143 & 0.0001 \\ 0.0005 & 0.0293 & 0.0001 & -0.0302 \end{pmatrix} & \begin{array}{c} A \\ C \\ G \\ T \end{array} \end{array}$$

whose negative trace provides the evolutionary distance $d = -tr[\Pi \log(\Pi^{-1} \mathbf{O}(t))] = 0.09152$.

The reader interested in more sophisticated applications of the GTR distance will find useful examples in Lanave et al. [48], Rodriguez et al. [63], and Cantanzaro et al. [10, 11].

8.3 The Least-Squares Paradigm of Phylogenetic Estimation

A paradigm of phylogenetic estimation is a quantitative criterion used to discern a phylogeny from among plausible alternatives. One of the earliest paradigms was introduced by Cavalli-Sforza and Edwards [15] and is known as *the additive model* or the *the least-squares model* of phylogenetic estimation [9].

Cavalli-Sforza and Edwards observed that as molecular data provide the most detailed anatomy possible for any organism, the diversity of life on Earth must be reflected in them. Hence, if evolution of a set of molecular data from taxa could be seen as a tree, then it could be described through a process that changes nucleotides over time. The trajectories described by such a process would split as taxa diverges, unite as taxa hybridize, end as taxa become extinct, and living taxa would be represented by the intercept of the process and the "now" plane (Fig. 8.2).

In general, we do not have a sampling of such a process over time but only the knowledge of the living taxa. Hence, in absence of further information, one may be able only to reconstruct the projection of the process onto the "now" plane rather than the process itself. Note that altough the evolutionary process over time is "directed," its projection is not (Fig. 8.2). Thus, when the projection is considered, the direction of evolution is definitely missed. Nevertheless, the projection of the evolutionary process constitutes still an important piece of information for the analyzed taxa; for this reason, Cavalli-Sforza and Edwards proposed a possible paradigm to recover it.

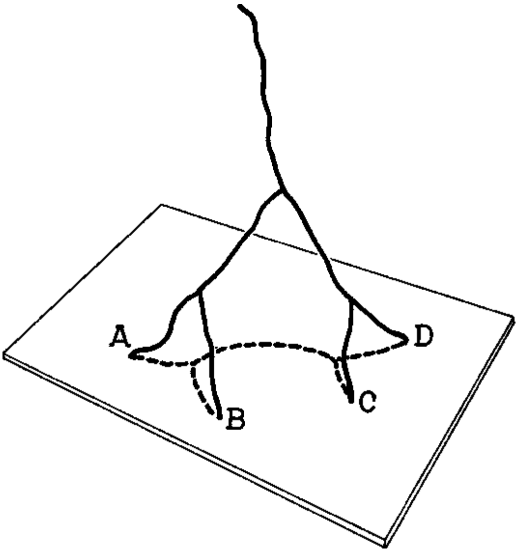


Fig. 8.2 An evolutionary process and its projection onto the "now" plane – from Cavalli-Sforza and Edwards [15]

The authors first considered the problem of how to represent formally a projection (phylogeny) of the evolutionary process. In order to remark the lack of a direction in evolution, the authors proposed to remove the root and the orientation in the edges of a phylogeny and represented it as an unrooted binary tree, i.e., an undirected acyclic graph in which each internal vertex has degree three. The degree constraint has not necessarily a biological foundation but helped the authors to formalize the evolutionary process. In fact, given n taxa, the degree constraint implies that the number of edges in a phylogeny T is $(2n-3)$ and the number of internal vertices is $(n-2)$. To prove the claim note that as T is a tree, it holds that:

$$|\mathcal{E}_i(T)| + |\mathcal{E}_e(T)| = |V_i| + |V_e| - 1, \tag{8.10}$$

where $\mathcal{E}_e(T)$ and $\mathcal{E}_i(T)$ are the set of external and internal edges of T, respectively. Moreover, since internal vertices have degree three, the following property holds:

$$2|\mathcal{E}_i(T)| + 2|\mathcal{E}_e(T)| = 3|V_i| + |V_e|. \tag{8.11}$$

Combining (8.10) and (8.11) it follows that $|V_i| = (n-2)$ and $|\mathcal{E}_i| = (n-3)$. Thus, a phylogeny $T \in \mathcal{T}$ can be seen as an unrooted binary tree in which the n taxa are the n leaves of T and the common ancestors are internal vertices of degree three. It is worth noting that dealing with unrooted binary trees does not introduces oversimplifications since it is easy to see that any m-ary tree can be transformed into a phylogeny by adding "dummy" vertices and edges (e.g., see Fig. 8.3).

Cavalli-Sforza and Edwards encoded a phylogeny in $\mathcal{T}$ by means of an *Edge–Path incidence matrix of a Tree* (EPT) (see [53, p. 550]) i.e., a network matrix $\mathbf{X}$ having a row for each path between two leaves and a column for each edge.

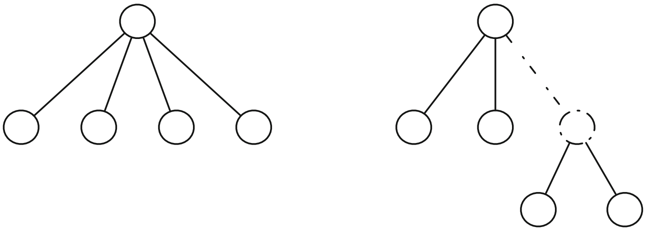
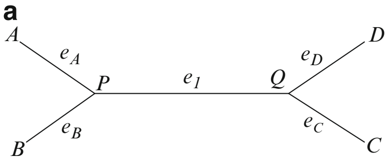


Fig. 8.3 The 4-ary tree (on the *left*) can be transformed into an unrooted binary tree by adding a dummy vertex and edge (*dashed*, on the *right*)



b

	e_A	e_B	e_C	e_D	e_1
Path AB	1	1	0	0	0
Path AC	1	0	1	0	1
Path AD	1	0	0	1	1
Path BC	0	1	1	0	1
Path BD	0	1	0	1	1
Path CD	0	0	1	1	0

Fig. 8.4 (**a**) An example of a phylogeny of four taxa (modeled as an unrooted binary tree in which each internal vertex has degree 3) and its associated EPT matrix (**b**)

The generic entry $x_{rs,e}$ of matrix $\mathbf{X}$ is equal to 1 if edge e belongs to the path p_{rs} from leaf r to leaf s and 0 otherwise. As an example, Fig. 8.4b shows the EPT matrix corresponding to the phylogeny shown in Fig. 8.4a. Hence, the authors proposed a model in which each evolutionary distance d_{rs}, $r, s \in \Gamma$, among pairwise molecular data could be thought of as the resulting sum of mutation events accumulated on edges belonging to the path p_{rs} linking taxa r and s on $\mathbf{X}$. In other words, fixed a phylogeny $\mathbf{X}$ and defined w_e as the amount of mutation events on edge e, Cavalli-Sforza and Edwards asserted that:

$$\mathbf{X}\mathbf{w} = \mathbf{D}^{\triangle}, \tag{8.12}$$

where $\mathbf{w} = \{w_e\}$ is the edge weight vector associated with $\mathbf{X}$, and $\mathbf{D}^{\triangle}$ is a $n(n-1)/2$ vector whose components are obtained by taking row by row the entries of the strictly upper triangular matrix $\mathbf{D} = \{d_{rs}\}$.

In general, for a fixed matrix $\mathbf{X}$, (8.12) may not admit solutions; for this reason, the authors proposed the use of the ordinary least-squares (OLS) to find the entries of vector $\mathbf{w}$. Specifically, the authors suggested that the values $\rho_{rs} = \sum_{e \in p_{rs}} x_{rs,e} w_e$ should minimize the function,

$$\sum_{r,s\in\Gamma:r\neq s}(d_{rs}-\rho_{rs})^2=\sum_{r,s\in\Gamma:r\neq s}\left(d_{rs}-\sum_{e\in p_{rs}}x_{rs,e}w_e\right)^2,$$

i.e., minimize the quadratic error related to the approximation of the evolutionary process with its projection. This condition holds when

$$\mathbf{w}=\mathbf{X}^{\dagger}\mathbf{D}^{\triangle},$$

where $\mathbf{X}^{\dagger}$ is the Moore–Penrose pseudo-inverse matrix of $\mathbf{X}$. Thus, Cavalli-Sforza and Edwards' paradigm of phylogenetic estimation may be stated in terms of the following NP-hard convex optimization problem [22]:

Problem 8.2. The ordinary least-squares problem (OLSP)

$$\min_{\mathbf{X}\in\mathcal{X},\mathbf{w}\in\mathbb{R}^{2n-3}}\quad f(\mathbf{X})=\sum_{r,s\in\Gamma:r\neq s}\left(d_{rs}-\sum_{e\in p_{rs}}x_{rs,e}w_e\right)^2,$$

where $\mathcal{X}$ denotes the set of all possible EPT matrices coding phylogenies. We refer the reader interested in a mathematical description of the necessary and sufficient conditions that characterize the set $\mathcal{X}$ to [14].

8.3.1 Modified Least-Squares Paradigms of Phylogenetic Estimation

A number of authors proposed some modifications to Cavalli-Sforza and Edwards' model. Specifically, Fitch and Margoliash [31] observed that OLSP implicitly considers the evolutionary distances d_{rs} among pairwise molecular data as uniformly distributed independent random variables, a hypothesis that cannot be considered generally true due to the common evolutionary history of the analyzed taxa and the presence of sampling errors in molecular data. Hence, Fitch and Margoliash proposed to modify Cavalli-Sforza and Edwards' paradigm by introducing the quantities ω_{rs} representing the variances of d_{rs}. They set $\omega_{rs}=1/d_{rs}^2$, $r,s\in\Gamma$, and stated the following paradigm:

Problem 8.3. The weighted least-squares problem (WLSP)

$$\min_{\mathbf{X}\in\mathcal{X},\mathbf{w}\in\mathbb{R}^{2n-3}}\quad f(\mathbf{X})=\sum_{r,s\in\Gamma:r\neq s}\omega_{rs}\left(d_{rs}-\sum_{e\in p_{rs}}x_{rs,e}w_e\right)^2.$$

Later, Chakraborty [16] and Hasegawa et al. [38] proposed a very similar paradigm, called the generalized least-squares problem (GLSP), in which the variances ω_{rs} are replaced by the covariances of d_{rs}. Nowadays, GLSP has fallen into disuse due to its statistical inconsistency problems [9].

8.3.2 Drawbacks of the Least-Squares Paradigms of Phylogenetic Estimation

Although the least-squares paradigm is a milestone in molecular phylogenetics, it is characterized by a number of drawbacks. For example, Cavalli-Sforza and Edwards' paradigm returns a *tree metric*, i.e., a phylogeny whose edge weights are non-negative [73, 80], whenever the distance matrix **D** satisfies the *ultrametric property*

$$d_{rs} \leq \max\{d_{rq}, d_{qs}\} \qquad r, s, q \in \Gamma \ : \ r \neq s \neq q$$

or the *additive property*

$$d_{rs} + d_{hk} \leq \max\{d_{rh} + d_{sk}, d_{rk} + d_{sh}\} \qquad r, s, h, k \in \Gamma \ : \ r \neq s \neq h \neq k.$$

Specifically, when **D** is ultrametric or additive, the solution of Problem 8.2 is unique and obtainable in polynomial time through the UPGMA greedy algorithm [74] or the sequential algorithm [80], respectively.

Unfortunately, when **D** is generic (e.g., when it is obtained by means of the THM model, see Sect. 8.2), the least-squares paradigm may lead to the occurrence of negative entries in the vector **w**, i.e., to a phylogeny that is not a tree metric [32, 47]. Negative edge weights are infeasible both from a conceptual point of view (a distance, being an expected number of mutation events over time, cannot be negative [45]) and from a biological point of view (evolution cannot proceed backwards [57, 77]). For the latter reason at least, non-tree metric phylogenies are generally not accepted in molecular phylogenetics [35].

In response, some authors investigated the consequences of adding or guaranteeing the positivity constraint of edge weights in the least-squares paradigm.

Gascuel and Levy [33] observed that the presence of the positivity constraint transforms any least-square model into a non-negative linear regression problem which involves projecting the distance matrix **D** onto the positive cone defined by the set of tree metrics (see also [5, p. 187]). Thus, the authors designed an iterative polynomial time algorithm able to generate a sequence of least-squares projections of **D** onto such a set until an additive distance matrix (and the corresponding phylogeny) is obtained.

Farach et al. [26] proposed an alternative approach to impose the positivity constraint. Specifically, the authors proposed to find the minimal perturbation of the distance matrix **D** that guarantees the satisfaction of the additive or the ultrametric property. Farach et al. [26] proposed the $\mathcal{L}_\infty$-norm and $\mathcal{L}_1$-norm to constraint the entries of **D** to satisfy the additive (ultrametric) property, and proved that such a problem can be solved in polynomial time when **D** is required to be ultrametric under the $\mathcal{L}_\infty$-norm. By contrast, the authors proved that their approaches become hard when an ultrametric or an additive distance matrix is required under the $\mathcal{L}_1$-norm.

Finally, Barthélemy and Guénoche [3] and Makarenkov and Leclerc [50] proposed a Lagrangian relaxation of the positivity constraint to guarantee metric trees. Both algorithms are iterative and apply to the OLSP and the WLSP, respectively. Specifically, starting from a leaf, the algorithms generate a phylogeny with a growing number of leaves by solving an optimization problem in which the best non-negative edge weights that minimize the OLSP (respectively the WLSP) are found. Both algorithms are polynomial time and characterized by a computational complexity of $O(n^4)$ and $O(n^5)$, respectively. FITCH, was also proposed by Felsenstein [27].

A second and possibly more serious drawback of the least-squares is the statistical inconsistency of some paradigms. Specifically, a part from the OLSP which proves to be statistically consistent [23, 68], the only case in which the WLSP is known to be consistent, is when the variances ω_{rs} are set to the inverse of the product of two strictly positive constants α_i and α_j. By contrast the GLSP is generally inconsistent [35].

8.4 The Minimum Evolution Paradigm of Phylogenetic Estimation

Kidd and Sgaramella-Zonta [45] and Beyer et al. [4] independently proposed an alternative paradigm known as *the minimum evolution problem* or *the minimum evolution paradigm* of phylogenetic estimation [9].

The minimum evolution paradigm arises from Cavalli-Sforza and Edwards' model but mainly differs for the way in which a phylogeny is chosen from among possible alternatives. In fact, the minimum evolution criterion states that if the evolutionary distances d_{rs} were unbiased estimates of the *true evolutionary distances* (i.e., the distances that one would obtain if all the molecular data from the analyzed taxa were available), then the true phylogeny would have an expected length shorter than any other possible phylogeny compatible with **D**. Hence, the minimum evolution paradigm aims at finding the phylogeny whose sum of edge weights, estimated from the corresponding evolutionary distances, is minimum [9].

It is worth noting that the minimum evolution criterion does not asses that molecular evolution follows minimum paths, but states, according to classical evolutionary theory, that a minimum length phylogeny may properly approximate the real phylogeny of well-conserved molecular data, i.e., data whose basic biochemical function has undergone small change throughout the evolution of the observed taxa [4]. That evolution proceeds by small rather than smallest changes is due to the fact that the neighborhood of possible alleles that are selected at each instant of the life of a taxon is finite, and perhaps more important, the selective forces acting on the taxon may not be constant throughout its evolution [4, 80]. Over the long term (periods of environmental change, including the intracellular environment), small changes will not generally provide the smallest change. Thus, a minimum length phylogeny provides a lower bound on the total number of mutation events that could have occurred along evolution of the observed taxa.

Different versions of the minimum evolution paradigm are discussed in the literature on phylogenetics, and each one is characterized by its own edge weight estimation model [9]. Specifically, we can distinguish between the least-squares edge weight estimation model [24, 68, 69] and the linear programming edge weight estimation model [4, 14, 80]. In the next sections, we shall analyze both families in detail.

8.4.1 The Minimum Evolution Paradigm Under the Least-Squares Edge Weight Estimation Model

The earliest minimum evolution paradigm of phylogenetic estimation was proposed by Kidd and Sgaramella-Zonta [45] and exploits Cavalli-Sforza and Edwards' model to estimate edge weights. The authors proposed to change the objective function of the OLSP with

$$f(\mathbf{X}) = \| \mathbf{w} \|_1 = \| \mathbf{X}^{\dagger}\mathbf{D}^{\triangle} \|_1 \tag{8.13}$$

giving rise to the following NP-hard convex optimization problem [9]:

Problem 8.4. The minimum evolution under least-squares problem (MELSP)

$$\min_{\mathbf{X}\in\mathcal{X}} \quad f(\mathbf{X}) = \| \mathbf{X}^{\dagger}\mathbf{D}^{\triangle} \|_1 \ .$$

Rzhetsky and Nei [68, 69] observed that the MELSP is statistically consistent, and such a property is also guaranteed when considering a relaxed version of the objective function in which edge weights are summed regardless their sign. However, Swofford et al. [77] criticized the choice of taking into account negative edge weights (or even their absolute value) in the objective function due to their biological unfeasibility. Thus, the authors proposed to replace the objective function (8.13) with

$$f(\mathbf{X}) = \sum_{e\in\mathcal{E}(T=\mathbf{X})|w_e\geq 0} w_e \, .$$

Gascuel et al. [35] investigated the statistical consistency of Swofford et al. [77] paradigm and obtained analogous results to Rzhetsky and Nei [68, 69]. At present, Swofford et al. [77] paradigm is one of the most used versions of minimum evolution, being implemented in the well-known software for phylogenetic estimation

"PAUP" [76]. The software is able to solve exactly instances of the paradigm containing upto 13 taxa and implements a hill-climbing metaheuristic to tackle larger instances of the problem.

Recently, Desper and Gascuel [24, 25] formalized the most recent version of the minimum evolution paradigm, called the Balanced Minimum Evolution problem (BME). The paradigm is based on Pauplin [59] seminal work in which the author criticized the biological consideration at the core of the OLSP. In fact, Pauplin noted that when computing the Moore-Penrose pseudo-inverse of the EPT matrix **X**, some edges can be weighted more than others. Since there is no biological justification for that, Pauplin proposed a new paradigm in which all edges of a phylogeny were weighted in the same way. The resulting objective function does not depend explicitly on edge weights and can be stated as follows:

$$f(T) = \sum_{r,s \in \Gamma : r \neq s} \frac{d_{rs}}{2^{\tau_{rs}}},$$

where τ_{rs} is called *the topological distance* and denotes the number of edges belonging to the path between taxa r and s in a phylogeny T [9]. Hence, BME can be stated in terms of the following optimization problem:

Problem 8.5. The Balanced Minimum Evolution Problem (BME)

$$\min_{T \in \mathcal{T}} \quad f(T) = \sum_{r,s \in \Gamma : r \neq s} \frac{d_{rs}}{2^{\tau_{rs}}}.$$

BME is known to be statistically consistent [24, 25] and its optimal solution satisfies the positivity constraint whenever the distance matrix satisfies the triangular inequality

$$d_{rs} \leq d_{rq} + d_{qs} \; \forall \; r, s, q \in \Gamma \; : \; r \neq s \neq q.$$

For the latter reason at least, finding the optimal solution to instances of BME is highly desirable. Unfortunately, this task seems hard, although at present no information about the complexity of BME is known in the literature.

Recent advances in the polyhedral combinatorics of BME led to solve exactly instances containing up to 20–25 taxa [13]. However, the size of the instances analyzable to the optimum is still far away from real needs; for this reason, the use of clustering heuristics (Fig. 8.5), such as the neighbor-joining tree (NJT) ([70, 75]), is common to tackle large instances of BME. Possibly, future developments on the polyhedral combinatorics of BME will provide fundamental new insights for the development of more efficient exact approaches to solution of the problem.

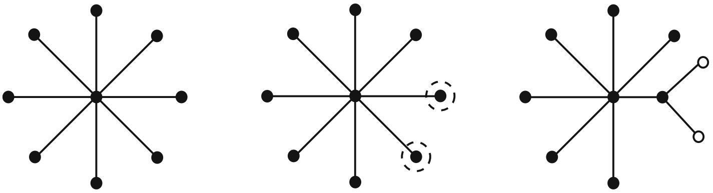


Fig. 8.5 Clustering heuristics: initially a graph-star is considered; subsequently two vertices (*circled*) are selected, marked (*white vertices*) and joined by an internal vertex. The algorithm is iterated on the remaining *black vertices* until a phylogeny is obtained

8.4.2 The Minimum Evolution Paradigm Under the Linear Programming Edge Weight Estimation Model

An alternative model to estimate edge weights in the minimum evolution paradigm is provided by linear programming. The model was introduced by Beyer et al. [4] and is based on the following motivation: if the evolutionary distances between pairs of molecular data have to reflect the number of mutation events required to convert one molecular sequence into another over time, then they must satisfy the triangle inequality. Moreover, since any edge weight of a phylogeny is de facto an evolutionary distance, also the entries of vector **w** must satisfy the triangle inequality. This last observation imposes that for each path p_{rs} from taxa r and s in **X**, the constraint $\sum_{e \in p_{rs}} w_e x_{rs,e} \geq d_{rs}$ is satisfied. Hence, Beyer et al. [4] proposed a possible paradigm of phylogenetic estimation consisting of solving the following mixed integer programming model:

Problem 8.6. The minimum evolution problem under linear programming (MELP)

$$\begin{aligned} \min_{\mathbf{X} \in \mathcal{X}, \mathbf{w} \in \mathbb{R}_{0+}^{2n-3}} \quad & f(\mathbf{X}, \mathbf{w}) = \| \mathbf{w} \|_1 \\ s.t. \quad & \mathbf{X}\mathbf{w} \geq \mathbf{D}^{\triangle}. \end{aligned}$$

MELP is a well-known APX-hard problem [26] for which the current exact algorithms described in the literature provide solutions to instances containing not more than a dozen taxa [14]. To the best of our knowledge, nothing is known about the statistical consistency of MELP.

8.4.3 Drawbacks of the Minimum Evolution Paradigm of Phylogenetic Estimation

There are mainly two drawbacks that affect the minimum evolution paradigm of phylogenetic estimation: the “rigidity” of its criterion and the hardness of its paradigms.

As regards to the first drawback, some authors, among which notably Felsenstein [29, p. 175], argued that the minimum evolution paradigms could prove unreliable as it neglects rate variation when estimating edge weights. This major criticism could be possibly overcome using non-homogeneous Markov models. Specifically, in a non-homogeneous Markov model, the Chapman–Kolmogorov master equation becomes [84]:

$$\dot{\mathbf{P}}(0,t) = \mathbf{R}(t)\mathbf{P}(0,t), \tag{8.14}$$

whose integral is given by

$$\mathbf{P}(0,t) = \mathbf{I} + \int_0^t \mathbf{R}(\tau)\mathbf{P}(0,\tau)\mathrm{d}\tau, \tag{8.15}$$

where **I** denotes the identity matrix. The use of the integral (8.15) could prove unpractical for an empirical use. However, note that (8.15) can be approximated through the Peano–Baker sequence

$$\begin{aligned} \mathbf{P}_0(0,t) &= \mathbf{I} \\ \mathbf{P}_k(0,t) &= \mathbf{I} + \int_0^t \mathbf{R}(\tau)\mathbf{P}_{k-1}(0,\tau)\mathrm{d}\tau, \;\; k = 1,2,\ldots \end{aligned} \tag{8.16}$$

since it is possible to prove that (8.16) converges to matrix $\mathbf{P}(0,t)$ when $k \to \infty$ [18]. Hence, under a non-homogeneous Markov model, the substitution probability matrix could be easily computed by means of iterative procedures that appropriately approximate (8.15).

Concerning the second drawback, it is easy to realize that the NP-hardness of the minimum evolution paradigms constitutes a big handicap for the development of exact solution approaches of practical use. Exact approaches are necessary to guarantee the optimality of a given solution and fundamental to investigate whether the hypotheses at the core of a criterion are well suited to describe the evolutionary process of the observed taxa. At present, most molecular datasets involve hundreds of taxa, whereas the current exact solution approaches have difficulty to tackle instances containing more than two dozen taxa (even smaller for the linear programming paradigm). Increasing the size of the datasets analyzable to the optimum is possibly one of the most challenging problems in molecular phylogenetics and warrants for sure further research efforts.

8.5 The Likelihood Paradigm of Phylogenetic Estimation

One of the most used criteria of phylogenetic estimation is the *likelihood criterion*. First formalized by Felsenstein [28], the likelihood criterion states that under many plausible explanations of an observed phenomenon, the one having the highest probability of occurring should be preferred to the others. When the likelihood criterion is applied to phylogenetic estimation, a phylogeny is defined to be optimal (or the most likely) if it has the highest probability of explaining the observed taxa. Thus, the likelihood paradigm consists of finding the phylogeny that maximizes a stochastic function, called *the likelihood function*, modeling a set of evolutionary hypotheses of the observed taxa.

The fundamental difference that distinguishes the likelihood paradigm from the least-squares and the minimum evolution paradigms is the nature of the information that it aims at finding. Specifically, if the least-squares and the minimum evolution paradigms aim at finding the best possible approximation of the projection of the evolutionary process of the observed taxa, the likelihood paradigm aims at reconstructing the most likely evolutionary process that originated the observed taxa. Hence, if the phylogeny of the least-squares and the minimum evolution paradigms is an unrooted binary tree, the phylogeny of the likelihood paradigm is a rooted phylogeny, i.e., full binary tree having $(2n-1)$ vertices.

Formally, the likelihood function is defined to be a recursive function of a fixed rooted phylogeny T, a model of molecular evolution M and an *observed data matrix* $\mathbf{S} = \{s_{rc}\}$, i.e., a matrix whose rth row represents the molecular sequence of the r-th taxon. Defined the quantity

$$L_c^r(i) = \begin{cases} 1, \text{ if } s_{rc} = i \\ 0, \text{ otherwise,} \end{cases}$$

for each leaf r of T, each column c of $\mathbf{S}$ and each $i \in \Upsilon$, and the quantity

$$L_c^v(i) = \left[\sum_{j \in \Upsilon} L_c^{v_1}(j) p_{ij}(t_{v_1,v}) \right] \left[\sum_{j \in \Upsilon} L_c^{v_2}(j) p_{ij}(t_{v_2,v}) \right],$$

for each internal vertex v of T having v_1 and v_2 as children, the likelihood function $L(T, \mathbf{S}, M)$ of T can be defined as

$$L(T, \mathbf{S}, M) = \prod_c \left[\sum_{j \in \Upsilon} L_c^\rho(j) \pi_j \right],$$

where ρ denotes the root of T. In the context of the likelihood paradigm, the expected numbers of substitutions per site t_{v_h,v_k} assume the analogous meaning of edge weights in the least-squares and minimum evolution paradigms. Hence, when a given model of molecular evolution is assumed to hold (e.g., the THM model), finding the most likely phylogeny for a set of molecular sequences means maximizing the nonlinear (usually) non-convex stochastic function $L(T, \mathbf{S}, M)$ over all the possible rooted phylogenies, and for each rooted phylogeny, over all the possible associated edge weights t_{v_h,v_k} and substitution probabilities $p_{ij}(t_{v_h,v_k})$.

The NP-hardness of the likelihood paradigm [62] justified the development of a number of approximate solution approaches typically based on hill climbing strategies. Specifically, the strategies consist of a first phase in which the structure of a best-so-far phylogeny is modified and a second phase in which the nonlinear optimization of edge weights and the substitution probabilities is performed. The two phases are consecutively iterated until a stopping criterion is satisfied (e.g., the number of iterations performed or the elapsed time) [7, 28, 64]. A systematic review of the hill climbing strategies for the likelihood paradigm is out of the scope of the present chapter and can be found in Bryant et al. [7].

Recent mathematical advances on the likelihood paradigm led to overcome several limitations of the initial Felsenstein's model, such as the absence of a rate variation among sites [81] and the absence of correlated evolution among sites [61]. Moreover, several progresses have been done concerning the analysis of its statistical consistency and its *idenfiability*, i.e., the study of the conditions under which the likelihood function is at least injective, an aspect markably related to its consistency [7]. The reader may find useful to refer to Gascuel [32] and Gascuel and Steel [34] for an overview of these aspects.

8.5.1 The Bayesian Paradigm of Phylogenetic Estimation

Given a dataset of molecular sequences, suppose we have sufficient empirical evidence to assert that the evolution of the observed taxa followed a specific stochastic process. Then, we could try to combine this a priori information with the likelihood function in order to bias the search of the most probable phylogeny through those solutions that fit the known evolutionary process. This idea is at the core of the most recent likelihood-derived paradigm of phylogenetic estimation, called the *bayesian paradigm*, and will be briefly described in this section.

Similar to the likelihood paradigm, the bayesian paradigm aims at finding the phylogeny that has the highest probability to recover the evolutionary process of the observed taxa. However, the selection of the most probable phylogeny is performed in light of the a priori information. Specifically, the a priori information is usually modeled by means of peculiar probability distributions, called *prior distributions*, which mainly concern three parameters, namely: the *topology*, i.e., the structure of the phylogeny, edge weights and the substitution probabilities. Defined

$$\Theta = \{t_{v_h, v_k} \in \mathbb{R}_{0+} : (v_k, v_k) \in T, \ \forall\, T \in \mathcal{T}\},$$

as the edge weight space and

$$\mathcal{R} = \left\{ p_{ij}(t) \in [0,1] : \sum_{j \in \Upsilon} p_{ij}(t) = 1, \ \forall\, i, j \in \Upsilon, \ t \in \mathbb{R}_{0+} \right\},$$

as the substitution probability space, the bayesian paradigm considers the prior distributions $\gamma(T)$, $\gamma(t)$, and $\gamma(R)$, to model the a priori information on $\mathcal{T}$, Θ, and $\mathcal{R}$, respectively. Selected an appropriate model of molecular evolution M, the prior distributions are then combined with the likelihood function to provide a *posterior density function* $B(T, \mathbf{S}, M)$ that represents the probability distribution of phylogenies conditional on the observed data matrix $\mathbf{S}$, the model M and the priors distributions $\gamma(T)$, $\gamma(t)$, and $\gamma(R)$. Maximizing $B(T, \mathbf{S}, M)$ is the goal of the bayesian paradigm.

According to Bayes' theorem, fixed a phylogeny T_i and denoted t_i and R_i the corresponding subspaces of edge weights and substitution probabilities, the mathematical expression of the posterior probability $B(T_i, \mathbf{S}, M)$ of T_i can be written as:

$$B(T_i, \mathbf{S}, M) = \frac{L_\int(T_i, \mathbf{S}, M)\gamma(T_i)}{\sum_{T_j \in \mathcal{T}} L_\int(T_j, \mathbf{S}, M)\gamma(T_j)}, \tag{8.17}$$

where $\gamma(T_i)$ denotes the prior probability of T_i, and $L_\int(T_i, \mathbf{S}, M)$ denotes the integral of the likelihood function $L(T_i, \mathbf{S}, M)$ over all possible edge weights and substitution probabilities [41], i.e.,

$$L_\int(T_i, \mathbf{S}, M) = \int_{t_i} \int_{R_i} L(T_i, \mathbf{S}, M)\gamma(t')\gamma(R')\mathrm{d}t'\mathrm{d}R'.$$

Hence, finding the optimal solution for the bayesian paradigm means finding the phylogeny T_i, the associated edge weights and the substitution probabilities that globally maximize the posterior probability distribution of phylogenies $B(T, \mathbf{S}, M)$. Since finding the maximum a posteriori phylogeny implicitly implies being able to solve the likelihood paradigm, solving the bayesian paradigm is NP-hard [29].

The recursive nature of the likelihood function and the intractability of computing the denominator of Bayes' theorem prevent an analytical approach to solution of the bayesian paradigm. Hence, the maximum a posteriori phylogeny is usually computed by means of a Markov chain Monte Carlo (MCMC) algorithm [30], i.e., an algorithm that samples $B(T, \mathbf{S}, M)$ through a stochastic generation of phylogenies in $\mathcal{T}$ ([49, 52, 83]). Sampling $B(T, \mathbf{S}, M)$ is extremely time consuming; therefore, the bayesian estimations may take even weeks [42]. However, as observed by Yang [82] and Huelsenbeck et al. [41, 43], the sampling process has also the indisputable benefit of providing a measure of the reliability of the best-so-far solution found. In fact, by sampling stochastically around the (best local) maximum a posteriori phylogeny T^*, the bayesian paradigm could determine support values for the subtrees of T^*, i.e., measures of the posterior probability that the subtrees are true.

The bayesian paradigm is possibly the most complex among the phylogenetic estimation paradigms currently available in the literature on molecular phylogenetics. The recent computational advances obtained by Ronquist and Huelsenbeck [65] speeded up the execution of the MCMC algorithm and widened the use of the bayesian paradigm. However, the lack of a systematic investigation of its statistical consistency and the unclear dependence of the posterior density function on the a priori information [82] possibly make the bayesian paradigm still unripe for phylogenetic estimation [1].

8.5.2 Drawbacks of the Likelihood and the Bayesian Paradigms of Phylogenetic Estimation

The higher the complexity of a paradigm, the higher the number of draw-backs that could arise, and the likelihood and the bayesian paradigms do not escape the rule. Specifically, a number of computational and theoretical drawbacks affect the two paradigms. The computational drawbacks mainly involve (i) the optimization aspects of the likelihood function and (ii) the sampling process in the bayesian paradigm. The theoretical drawbacks concern the evolutionary hypotheses at the core of the likelihood and bayesian criteria.

As regards to the computational drawbacks, in Sect. 8.5 we have seen that finding the most likely phylogeny for a set of taxa involves maximizing a nonlinear and generally non-convex stochastic function over all the possible phylogenies in $\mathcal{T}$, and for each phylogeny, over all the possible edge weights and substitution probabilities. Notoriously, this task can be only performed in an approximate way, due to a lack of general mathematical conditions that guarantee the global optimality of a solution in nonlinear non-convex programming [21, 54]. Hence, although it is possible (at least for small datasets) to enumerate all the possible phylogenies in $\mathcal{T}$, it is not possible to optimize globally edge weights and the substitution probabilities of a fixed phylogeny T. This fact may affect negatively the statistical consistency of the likelihood and the bayesian paradigms. In fact, the local optima of the likelihood function grows up exponentially in function of the number of taxa considered [7, 19, 20]. Thus, fixed a phylogeny T, the global optimum of the likelihood function is generally approximated by means of hill-climbing techniques that jump from local optimum to another one until a stopping criterion is satisfied (e.g., the number of iterations performed or the elapsed time) [7,28,64]. Assume that two phylogenies T_1 and T_2 are given, and let μ_1 and μ_2 be two vectors whose entries are edge weights and the substitution probabilities associated to T_1 and T_2, respectively. Let z_1 and z_2, the likelihood values of T_1 and T_2 for μ_1 and μ_2, respectively, and assume, without loss of generality, that $z_1 > z_2$. Due to the local nature of the optima μ_1 and μ_2, there could exists another local optimum, say $\hat{\mu}_2$, such that $\hat{z}_2 > z_1 > z_2$. If the hill-climbing algorithm finds $\hat{\mu}_2$ before μ_2, then we will consider T_2 as a better phylogeny than T_1, otherwise we will discard T_2 in favor of T_1. Hence, it is easy to realize that if one of the two phylogenies is the true phylogeny, its acceptance is subordinated to the goodness of the hill-climbing algorithm used to optimize the likelihood function, and as a result the statistical consistency of the likelihood and bayesian paradigms may be seriously compromised.

Some authors argued that multiple local optima should arise infrequently in real datasets [64], but this conjecture was proved false by Bryant et at. [7] and

Catanzaro et al. [12]. Specifically, Bryant et al. [7] observed that changing the model of molecular evolution influences the presence of multiple optima in the likelihood function, and Catanzaro et al. [12] showed a number of real datasets affected by strong multimodality of the likelihood function. Despite the importance of the topic, to the best of our knowledge nobody was able to propose a plausible solution to this critical aspect.

A second computational drawback concerns the sampling process of the bayesian paradigm. In fact, as shown in Sect. 8.5.1, the approximation of the posterior density function is generally performed by means of a MCMC algorithm (e.g., the Metropolis or the Gibbs sampling algorithm [30]) that performs random walks in $\mathcal{T}$. The random walk should be sufficiently diversified to sample potentially the whole $\mathcal{T}$ and avoid double backs (i.e., to sample phylogenies already visited). Unfortunately, despite the recent computational advances in the bayesian paradigm [65], no technique may guarantee a sufficient diversification of the sampling process. Hence, the convergence to the maximum a posteriori phylogeny in practice becomes the convergence to the best-so-far a posteriori phylogeny that can be arbitrarily distinct from the true phylogeny (see [29, p. 296]).

As regards to the theoretical drawbacks, it is worth noting that the evolutionary hypotheses at the core of the likelihood and bayesian criteria of phylogenetic estimation are at the same time their strength and their weakness. For example, if a proposed model of molecular evolution matches (at least roughly) the real evolutionary process of a set of molecular data, then the likelihood and the bayesian paradigms could succeed in recovering the real phylogeny of the corresponding set of taxa (provided a solution to their computational drawbacks). However, if it is not the case, the paradigms will just provide a (sub)optimal solution for that model that may completely mismatch the real phylogeny. This aspect becomes evident e.g., in Rydin and Källersjö [67]'s article where, for a same dataset, two different Markov model of molecular evolution are used and two different maximum posterior phylogenies are obtained both having the 100% posterior probability of supporting the true phylogeny. concerns in general all the paradigms discussed in this chapter and possibly there is no easy solution for it.

Finally, a second theoretical drawback concerns the prior distributions of the bayesian paradigm. In fact, it is worth noting that if on one hand a strength of the bayesian paradigm is the ability to incorporate the a priori information, on the other hand this information is rarely available, hence in practical applications the prior distributions are generally modeled as uniform distributions, frustrating the potential strengths of the paradigm [1]. Moreover, it is unclear what type of information is well suited for a prior distribution; how possible conflicts among different sets of a priori information can be resolved; and if the inclusion of prior distributions strongly bias the estimation process. Huelsenbeck et al. [43] vaguely claimed "in a typical Bayesian analysis of phylogeny, the results are likely to be rather insensitive to the prior," but this results was not confirmed by Yang [82] who observed that "[...] the posterior probabilities of trees vary widely over simulated datasets [...] and can be unduly influenced by the prior [...]." Possibly, further research efforts are needed to provide answers to these practical concerns.

8.6 Conclusion

The success of a criterion of phylogenetic estimation is undoubtedly influenced by the quality of the evolutionary hypotheses at its core. If the hypotheses match (at least roughly) the real evolutionary process of a set of taxa, then the criterion will hopefully succeed in recovering the real phylogeny. Otherwise, the criterion will miserably fail, by suggesting an optimal phylogeny that mismatch partially or totally the correct result. Since we are far away from a complete understanding of the complex facets of evolution, it is not generally possible to assess the superiority of a criterion over others. Hence, families of estimation criteria cohabit in the literature of molecular phylogenetics, by providing different perspectives about the evolutionary process of the involved taxa.

In this chapter, we have presented a general introduction of the existing literature about molecular phylogenetics. Our purpose has been to introduce a classification scheme in order to provide a general framework for papers appearing in this area. In particular, three main criteria of phylogenetic estimation have been outlined, the first based on the least-squares paradigm, first proposed by Cavalli-Sforza and Edwards [15], the second based on the minimum evolution paradigm, independently proposed by Kidd and Sgaramella-Zonta [45] and Beyer et al. [4], and the third based on the likelihood paradigm, first proposed by Felsenstein [28]. This division has been further disaggregated into different, approximately homogeneous sub-areas, and the basic aspects of each have been pointed out. For each, also, the most relevant issues affecting their use in tackling real-world sized problems have been outlined, as have the most interesting refinements deserving further research effort.

Acknowledgment Daniele Catanzaro acknowledges support from the Belgian National Fund for Scientific Research (F.N.R.S.) of which he is “Chargé de Recherches.” Raffaele Pesenti and the anonymous reviewers for their valuable comments on previous versions of the manuscript. Finally, thanks to Prof. Mike Steel and Dr. Rosa Maria Lo Presti for helpful and exciting discussions.

References

1. J. K. Archibald, M. E. Mort, and D. J. Crawford. Bayesian inference of phylogeny: A nontechnical primer. *Taxon*, 52:187–191, 2003
2. D. A. Bader, B. M. E. Moret, and L. Vawter. Industrial applications of high-performance computing for phylogeny reconstruction. In *SPIE ITCom: Commercial application for high-performance computing*, pages 159–168. SPIE, WA, 2001
3. J. P. Barthélemy and A. Guénoche. *Trees and proximity representations*. Wiley, NY, 1991
4. W. A. Beyer, M. Stein, T. Smith, and S. Ulam. A molecular sequence metric and evolutionary trees. *Mathematical Biosciences*, 19:9–25, 1974
5. Å. Björck. *Numerical methods for least-squares problems*. SIAM, PA, 1996
6. J. Brinkhuis and V. Tikhomirov. *Optimization: Insights and applications*. Princeton University Press, NJ, 2005
7. D. Bryant, N. Galtier, and M. A. Poursat. Likelihood calculation in molecular phylogenetics. In O. Gascuel, editor, *Mathematics of evolution and phylogeny*. Oxford University Press, NY, 2005

8. R. M. Bush, C. A. Bender, K. Subbarao, N. J. Cox, and W. M. Fitch. Predicting the evolution of human influenza A. *Science*, 286(5446):1921–1925, 1999
9. D. Catanzaro. The minimum evolution problem: Overview and classification. *Networks*, 53(2): 112–125, 2009
10. D. Catanzaro, L. Gatto, and M. Milinkovitch. Assessing the applicability of the GTR nucleotide substitution model through simulations. *Evolutionary Bioinformatics*, 2:145–155, 2006
11. D. Catanzaro, R. Pesenti, and M. Milinkovitch. A non-linear optimization procedure to estimate distances and instantaneous substitution rate matrices under the GTR model. *Bioinformatics*, 22(6):708–715, 2006
12. D. Catanzaro, R. Pesenti, and M. C. Milinkovitch. A very large-scale neighborhood search to estimate phylogenies under the maximum likelihood criterion. Technical report, G.O.M. – Computer Science Department – Université Libre de Bruxelles (U.L.B.), 2007
13. D. Catanzaro, M. Labbé, R. Pesenti, and J. J. Salazar-Gonzalez. The balanced minimum evolution problem. Technical report, G.O.M. – Computer Science Department – Université Libre de Bruxelles (U.L.B.), 2009
14. D. Catanzaro, M. Labbé, R. Pesenti, and J. J. Salazar-Gonzalez. Mathematical models to reconstruct phylogenetic trees under the minimum evolution criterion. *Networks*, 53(2):126–140, 2009
15. L. L. Cavalli-Sforza and A. W. F. Edwards. Phylogenetic analysis: Models and estimation procedures. *American Journal of Human Genetics*, 19:233–257, 1967
16. R. Chakraborty. Estimation of time of divergence from phylogenetic studies. *Canadian Journal of Genetics and Cytology*, 19:217–223, 1977
17. B. S. W. Chang and M. J. Donoghue. Recreating ancestral proteins. *Trends in Ecology and Evolution*, 15(3):109–114, 2000
18. L. Chisci. *Sistemi Dinamici – Parte I.* Pitagora, Italy, 2001
19. B. Chor, M. D. Hendy, B. R. Holland, and D. Penny. Multiple maxima of likelihood in phylogenetic trees: An analytic approach. *Molecular Biology and Evolution*, 17(10):1529–1541, 2000
20. B. Chor, M. D. Hendy, and S. Snir. Maximum likelihood jukes-cantor triplets: Analytic solutions. *Molecular Biology and Evolution*, 23(3):626–632, 2005
21. A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. SIAM, PA, 2000
22. W. H. E. Day. Computational complexity of inferring phylogenies from dissimilarity matrices. *Bulletin of Mathematical Biology*, 49:461–467, 1987
23. F. Denis and O. Gascuel. On the consistency of the minimum evolution principle of phylogenetic inference. *Discrete Applied Mathematics*, 127:66–77, 2003
24. R. Desper and O. Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum evolution principle. *Journal of Computational Biology*, 9(5):687–705, 2002
25. R. Desper and O. Gascuel. Theoretical foundations of the balanced minimum evolution method of phylogenetic inference and its relationship to the weighted least-squares tree fitting. *Molecular Biology and Evolution*, 21(3):587–598, 2004
26. M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13:155–179, 1995
27. J. Felsenstein. An alternating least-squares approach to inferring phylogenies from pairwise distances. *Systematic Biology*, 46:101–111, 1997
28. J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981
29. J. Felsenstein. *Inferring phylogenies*. Sinauer Associates, MA, 2004
30. G. S. Fishman. *Monte Carlo: Concepts, algorithms, and applications*. Springer, NY, 1996
31. W. M. Fitch and E. Margoliash. Construction of phylogenetic trees. *Science*, 155:279–284, 1967
32. O. Gascuel. *Mathematics of evolution and phylogeny*. Oxford University Press, NY, 2005
33. O. Gascuel and D. Levy. A reduction algorithm for approximating a (non-metric) dissimilarity by a tree distance. *Journal of Classification*, 13:129–155, 1996
34. O. Gascuel and M. A. Steel. *Reconstructing evolution*. Oxford University Press, NY, 2007

35. O. Gascuel, D. Bryant, and F. Denis. Strengths and limitations of the minimum evolution principle. *Systematic Biology*, 50:621–627, 2001
36. P. H. Harvey, A. J. L. Brown, J. M. Smith, and S. Nee. *New uses for new phylogenies*. Oxford University Press, Oxford, 1996
37. M. Hasegawa, H. Kishino, and T. Yano. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981
38. M. Hasegawa, H. Kishino, and T. Yano. Dating the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22:160–174, 1985
39. D. P. Heyman and M. J. Sobel, editors. Stochastic models, volume 2 of *Handbooks in operations research and management science*. North-Holland, Amsterdam, 1990
40. S. Horai, Y. Sattah, K. Hayasaka, R. Kondo, T. Inoue, T. Ishida, S. Hayashi, and N. Takahata. Man's place in the hominoidea revealed by mitochondrial DNA genealogy. *Journal of Molecular Evolution*, 35:32–43, 1992
41. J. P. Huelsenbeck, B. Larget, P. van der Mark, and F. Ronquist. MrBayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001
42. J. P. Huelsenbeck, F. Ronquist, R. Nielsen, and J. P. Bollback. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294:2310–2314, 2001
43. J. P. Huelsenbeck, B. Larget, R. E. Miller, and F. Ronquist. Potential applications and pitfalls of bayesian inference of phylogeny. *Systematic Biology*, 51:673–688, 2002
44. T. H. Jukes and C.R. Cantor. Evolution of protein molecules. In H. N. Munro, editor, *Mammalian protein metabolism*, pages 21–123. Academic Press, NY, 1969
45. K. K. Kidd and L. A. Sgaramella-Zonta. Phylogenetic analysis: Concepts and methods. *American Journal of Human Genetics*, 23:235–252, 1971
46. M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16:111–120, 1980
47. M. K. Kuhner and J. Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal rates. *Molecular Biology and Evolution*, 11(3):584–593, 1994
48. C. Lanave, G. Preparata, C. Saccone, and G. Serio. A new method for calculating evolutionary substitution rates. *Journal of Molecular Evolution*, 20:86–93, 1984
49. S. Li, D. Pearl, and H. Doss. Phylogenetic tree construction using Markov chain Monte Carlo. *Journal of the American Statistical Association*, 95:493–508, 2000
50. V. Makarenkov and B. Leclerc. An algorithm for the fitting of a tree metric according to a weighted least-squares criterion. *Journal of Classification*, 16:3–26, 1999
51. M. A. Marra, S. J. Jones, C. R. Astell, R. A. Holt, A. Brooks-Wilson, Y. S. Butterfield, J. Khattra, J. K. Asano, S. A. Barber, S. Y. Chan, A. Cloutier, S. M. Coughlin, D. Freeman, N. Girn, O. L. Griffith, S. R. Leach, M. Mayo, H. McDonald, S. B. Montgomery, P. K. Pandoh, A. S. Petrescu, A. G. Robertson, J. E. Schein, A. Siddiqui, D. E. Smailus, J. M. Stott, G. S. Yang, F. Plummer, A. Andonov, H. Artsob, N. Bastien, K. Bernard, T. F. Booth, D. Bowness, M. Czub, M. Drebot, L. Fernando, R. Flick, M. Garbutt, M. Gray, A. Grolla, S. Jones, H. Feldmann, A. Meyers, A. Kabani, Y. Li, S. Normand, U. Stroher, G. A. Tipples, S. Tyler, R. Vogrig, D. Ward, B. Watson, R. C. Brunham, M. Krajden, M. Petric, D. M. Skowronski, C. Upton, and R. L. Roper. The genome sequence of the SARS-associated coronavirus. *Science*, 300(5624):1399–1404, 2003
52. B. Mau and M. A. Newton. Phylogenetic inference for binary data on dendograms using Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, 6:122–131, 1997
53. G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience, NY, 1999
54. G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Tod, editors. Optimization, volume 1 of *Handbooks in operations research and management science*. North-Holland, Amsterdam, 1989
55. C. Y. Ou, C. A. Ciesielski, G. Myers, C. I. Bandea, C. C. Luo, B. T. M. Korber, J. I. Mullins, G. Schochetman, R. L. Berkelman, A. N. Economou, J. J. Witte, L. J. Furman, G. A. Satten,

K. A. Maclnnes, J. W. Curran, and H. W. Jaffe. Molecular epidemiology of HIV transmission in a dental practice. *Science*, 256(5060):1165–1171, 1992

56. L. Pachter and B. Sturmfels. The mathematics of phylogenomics. *SIAM Review*, 49(1):3–31, 2007
57. R. D. M. Page and E. C. Holmes. *Molecular evolution: A phylogenetic approach*. Blackwell Science, Oxford, 1998
58. J. M. Park and M. W. Deem. Phase diagrams of quasispecies theory with recombination and horizontal gene transfer. *Physical Review Letters*, 98:058101–058104, 2007
59. Y. Pauplin. Direct calculation of a tree length using a distance matrix. *Journal of Molecular Evolution*, 51:41–47, 2000
60. P. A. Pevzner. *Computational molecular biology*. MIT, MA, 2000
61. D. D. Pollock, W. R. Taylor, and N. Goldman. Coevolving protein residues: Maximum likelihood identification and relationship to structure. *Journal of Molecular Biology*, 287(1): 187–198, 1999
62. S. Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(1):92–94, 2006
63. F. Rodriguez, J. L. Oliver, A. Marin, and J. R. Medina. The general stochastic model of nucleotide substitution. *Journal of Theoretical Biology*, 142:485–501, 1990
64. J. S. Rogers and D. Swofford. Multiple local maxima for likelihoods of phylogenetic trees from nucleotide sequences. *Molecular Biology and Evolution*, 16:1079–1085, 1999
65. F. Ronquist and J. P. Huelsenbeck. MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1572–1574, 2003
66. H. A. Ross and A. G. Rodrigo. Immune-mediated positive selection drives human immunodeficency virus type 1 molecular variation and predicts disease duration. *Journal of Virology*, 76(22):11715–11720, 2002
67. C. Rydin and M. Källersjö. Taxon sampling and seed plant phylogeny. *Cladistics*, 18:485–513, 2002
68. A. Rzhetsky and M. Nei. Theoretical foundations of the minimum evolution method of phylogenetic inference. *Molecular Biology and Evolution*, 10:1073–1095, 1993
69. A. Rzhetsky and M. Nei. Statistical properties of the ordinary least-squares generalized least-squares and minimum evolution methods of phylogenetic inference. *Journal of Molecular Evolution*, 35:367–375, 1992
70. N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987
71. E. Schadt and K. Lange. Codon and rate variation models in molecular phylogeny. *Molecular Biology and Evolution*, 19(9):1534–1549, 2002
72. E. Schadt and K. Lange. Applications of codon and rate variation models in molecular phylogeny. *Molecular Biology and Evolution*, 19(9):1550–1562, 2002
73. C. Semple and M. A. Steel. *Phylogenetics*. Oxford University Press, NY, 2003
74. P. H. A. Sneath and R. R. Sokal. *Numerical taxonomy*. W. K. Freeman and Company, CA, 1963
75. J. A. Studier and K. J. Keppler. A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution*, 5:729–731, 1988
76. D. L. Swofford. *PAUP* version 4.0*. Sinauer Associates, MA, 1997
77. D. L. Swofford, G. J. Olsen, P. J. Waddell, and D. M. Hillis. Phylogenetic inference. In D. M. Hillis, C. Moritz, and B. K. Mable, editors, *Molecular systematics*, pages 407–514. Sinauer Associates, MA, 1996
78. K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Molecular Biology and Evolution*, 10(3):512–526, 1993
79. P. J. Waddell and M. A. Steel. General time-reversible distances with unequal rates across sites: Mixing gamma and inverse gaussian distributions with invariant sites. *Molecular Phylogenetics and Evolution*, 8:398–414, 1997
80. M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer. Additive evolutionary trees. *Journal of Theoretical Biology*, 64:199–213, 1977

81. Z. Yang. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular Evolution*, 39:306–314, 1994
82. Z. Yang. Bayesian inference in molecular phylogenetics. In O. Gascuel, editor, *Mathematics of evolution and phylogeny*. Oxford University Press, NY, 2005
83. Z. Yang and B. Rannala. Bayesian phylogenetic inference using DNA sequences: A Markov chain Monte Carlo method. *Molecular Biology and Evolution*, 14:717–724, 1997
84. L. A. Zadeh and C. A. Desoer. *Linear system theory*. McGraw-Hill, NY, 1963

Chapter 9
Population Stratification Analysis in Genome-Wide Association Studies

Erika Salvi, Alessandro Orro, Guia Guffanti, Sara Lupoli, Federica Torri, Cristina Barlassina, Steven Potkin, Daniele Cusi, Fabio Macciardi, and Luciano Milanesi

Abstract Differences in genetic background within two or more populations are an important cause of disturbance in case–control association studies. In fact, when mixing together populations of different ethnic groups, different allele frequencies between case and control samples could be due to the ancestry rather than a real association with the disease under study. This can easily lead to a large amount of false positive and negative results in association study analysis. Moreover, the growing need to put together several data sets coming from different studies in order to increase the statistical power of the analysis makes this problem particularly important in recent statistical genetics research. To overcome these problems, different correction strategies have been proposed, but currently there is no consensus about a common powerful strategy to adjust for population stratification. In this chapter, we discuss the state-of-the-art of strategies used for correcting the statistics for genome-wide association analysis by taking into account the ancestral structure of the population. After a short review of the most important methods and tools available, we will show the results obtained in two real data sets and discuss them in terms of advantages and disadvantages of each algorithm.

9.1 Introduction

A genome-wide association study (GWAS) is defined as an examination of genetic variation across the human genome aimed to identify genetic associations with observable traits or qualitative dichotomous traits. To date, several genome-wide association studies have been performed to identify chromosomal regions containing disease-susceptibility loci by detecting differences in allele frequencies between affected (cases) and unaffected individuals (controls). Mapping genetic loci with GWAS is based on linkage disequilibrium (LD), which is defined as a

L. Milanesi (✉)
CNR – Institute for Biomedical Technologies, Via Fratelli Cervi 93, 20090 Segrate, MI, Italy
e-mail: luciano.milanesi@itb.cnr.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_9,

condition in which some combinations of alleles or genetic markers occur more or less frequently than can be accounted by chance. LD indicates that alleles at different loci on the same DNA strand are transmitted together. Leveraging on specific "properties" of the single nucleotide polymorphisms (SNPs), like their high allelic frequency and their unique position within the human genome, SNPs have been shown to act as universal markers able to flag genes and/or chromosomal regions potentially relevant for the disease under investigation. When a given SNP – or a cluster of SNPs – shows a statistically significant difference in their allelic or genotypic frequency between cases and controls, this finding points to a role of the locus mapped by those SNPs in the etiopathogenesis of the disease. If genetic variations are more frequent in subjects with the disease, the variations are said to be positively "associated," representing risk factors to develop the disease. The associated genetic variations are then considered pointers to the region of the human genome where the hypothetical disease-causing locus resides [1–4]. In case–control association studies, population stratification (PS) occurs when allele frequencies differ between cases and controls due to ancestry differences, ethnic background or even to "hidden" stratification. Population structure can lead to spurious findings between a phenotype and unlinked candidate loci, causing either false positive or false negative results when analysing SNPs for association [5–7]. To control these issues, different strategies have been proposed. Fst [8] and STRUCTURE [9] methods allow only detecting but not correcting the possible population substructure. Fst test measures the population genetic differentiation and assesses the variation in the subpopulations by quantifying the loss of heterozygosity. Fst strongly depends on the number of SNPs used. STRUCTURE assigns subjects to discrete subpopulations computing the likelihood a given genotype originated in each population. The major limitations of STRUCTURE are the intensive computational cost on large data sets if applied to genome-wide data sets and the sensitivity to the number of clusters defined by users before analysis. Recently, Li et al. [10] proposed a likelihood based algorithm that can substantially speed-up the calculations. Genomic control (GC) [11], EIGENSTRAT, based on principal-component analysis (PCA) [12] and Cochran–Mantel–Haenszel test (CMH) in PLINK [13] are the currently most common approaches used to correct PS in genetic association analysis. GC rescales the association statistics by a common overall factor λ (inflation factor) at each marker, while EIGENSTRAT and PLINK use multivariate techniques designed to reduce the data to a small number of dimensions, taking into account as much variability as possible. These methods enable explicit detection and correction of PS on a genome-wide scale. Limitations of GC depend on the uniform adjustment that may be insufficient because it is not specific for each marker and for the related allele frequency across populations. The threshold of inflation factor that allows considering a sample as sub-structured and the resulting association inflated by stratification is, however, not universally defined. PCA appears to be widely used with STRUCTURE to analyse the population structure of different worldwide populations as reported by Bauchet et al. [14], Tian et al. [15] and Price et al. [16] in European and European-American populations. In several simulation study, the PCA method was also the most powerful to control stratification effects using PCs as covariates in a logistic regression model [12, 17–19]. Among these methods based on PCA, EIGENSTRAT is the most widely used in GWAS [20–22]. Different studies proposed alternative methods or integration to PCA [23–25]. These different methods can use genotype information from a whole genome set of SNPs [16, 19–22] or from a set of selected informative markers (AIMs) [14, 15, 26, 27]. AIMs are defined as markers that show larger allele frequency differences between ancestral populations. There are currently several existing AIMs panels [5] that can be genotyped to estimate genetic ancestry, but all panels are not equivalent and may also not be robust enough in genetic studies with populations of unknown origins. In summary, there is currently no definitive consensus about a common strategy to adjust for PS. The goal of scientists studying genetic associations with a complex disease in samples that may include different (ethnic) groups is to examine the population ethnic background and to correct for stratification to find the genetic variants really relevant for the disease by avoiding false-positive associations. This chapter shows some examples looking into PS applying different methods to find the most efficient strategy to correct the observed findings based on the detected genetic variance in two association studies on schizophrenia. We assessed PS in the UCI sample composed by about 200 subjects and in the available CATIE-NIMH sample [28], made up of 1,492 individuals. Both samples are composed of different ethnic groups.

9.2 Materials and Methods

9.2.1 Subject Collection and Genotyping

We performed case–control association study in two different samples with different sizes: the UCI sample and the CATIE-NIMH sample.

9.2.1.1 UCI Sample

We studied 107 patients with chronic schizophrenia who have been recruited at the University of California at Irvine (UCI). Schizophrenia was diagnosed according to the criteria of the diagnostic and statistical manual of mental disorders (DSM) IV using structured clinical interviews. In addition, 91 healthy controls matched for age and gender have also been recruited at the UCI without any mental disease according to DSM IV. Written informed consent was obtained and blood samples collected from each patient and control. Eighty-four per cent of the individuals were Caucasians, 4.5% were Asians and the remaining 11.5% were African-Americans. Ethnic status was assessed by the clinician according to information about the place of birth of each individual together with that of their parents and grandparents, as well as their mother language. Genomic DNA was extracted from blood by a standard procedure. Genotyping was performed at the Department of Science and

Biomedical Technologies, University of Milan. For the genome-wide association study, approximately 750 ng of genomic DNA was used to genotype each subject for 317503 Phase I Hap Map tagging SNPs on the Infinium HumanHap300 BeadArrays (Illumina, San Diego, USA). Samples were processed according to the Illumina Infinium 2 assay. Briefly, each sample was whole-genome amplified, fragmented, precipitated and hybridized overnight for a minimum of 16 h at 48°C to locus-specific probes on the BeadArray. Non-specifically hybridized fragments were removed by washing while the remaining specifically hybridized DNA fragments were processed for the single base extension reaction, stained and imaged on an Illumina BeadArray Reader. Normalized bead intensity data obtained for each sample were analysed with Illumina Beadstudio 2.0 software, which generated SNP genotypes from fluorescent intensities using the manufacturer default cluster settings [29, 30]. After removal of SNPs with no calls and those with a minor allele frequency less than 0.01, we were left with 297197 SNPs with an average call frequency rate of 98.9%.

9.2.1.2 CATIE-NIMH Sample

The CATIE-NIMH sample contains 741 schizophrenics of the CATIE project matched with 751 controls collected from the NIMH Genetics repository 25, whose genotype and phenotype data are available to the scientific community (www.nimhgenetics.org). GWAS genotyping was conducted by Perlegen Sciences using the Affymetrix 500K "A" chipset (Nsp and Sty) and Perlegen custom 164K chip: each subject was genotyped for 495172 SNPs. In terms of ethnicity, 56.17% of subjects are Europeans, 29.62% Africans and 14.21% are selected as other or more than one racial category (American-Indian/Alaska Native, Asian, Black/African-American, Native Hawaiian/other Pacific Islander, White and Hispanic/Latino).

9.2.2 Stratification Approaches

To detect the possible PS, we applied different methods listed in Table 9.1 where for each method, the number of markers used, the statistical method on which are

Table 9.1 Methods for population homogeneity test

Software	# markers	Statistical method	Detection/correction
STRUCTURE	Limited	Bayesian approach	Detection
FST	Limited/GW	F statistics	Detection
Genomic control	Limited/GW	T statistics	Both
PLINK	Genome-wide	Cochran-Mantel-Haenszel	Both
EIGENSTRAT	Genome-wide	Armitage trend test	Both

GW genome-wide

based and their ability to perform detection or correction of PS, are shown. Fst and STRUCTURE methods allow only detecting but not correcting the possible population substructure using a limited number of SNPs. GC, EIGENSTRAT, based on PCA and CMH in PLINK are the currently most common approaches used to correct for PS in genetic association analysis using a whole genome set of SNPs.

From the entire genome-wide SNP panel, we selected a set of 400 SNPs that are unlinked to each other, not in regions of susceptibility to the disease of interest and equally distributed across the genome and we applied Fst and STRUCTURE methods.

9.2.2.1 FST

In population genetics, F-statistics describes the level of heterozygosity in a population, more specifically the degree of a reduction in heterozygosity when compared to Hardy–Weinberg expectation. F-statistics is defined by the ratio between observed and expected value of heterozygous genotypes.

$$F = 1 - \frac{O(f(Aa))}{E(f(Aa)} = 1 - \frac{O(f(Aa))}{2nq}, \tag{9.1}$$

where n is the frequency of major allele (A), q is the frequency of minor allele (a) and the expected value is calculated at the Hardy–Weinberg equilibrium.

The F-statistics can be partitioned in two terms that are related to different levels of population structure: F_{IS} correlates the heterozygosity of individuals in the subpopulations, whereas F_{ST} correlates the heterozygosity of subpopulation respect to the total population. In order to identify the genetic diversity due to allele frequency differences among population, we are interested in the F_{ST} term only. Fst test [8] calculates the gene frequency variation in the subpopulation relative to that in the total population by quantifying the loss of heterozygosity due to the existence of a population structure.

$$F_{\mathrm{ST}} = \frac{H_{\mathrm{T}} - H_{\mathrm{S}}}{H_{\mathrm{T}}}, \tag{9.2}$$

where H_{T} is total expected heterozygosity and H_{S} is the average heterozygosity derived from each subpopulation. The value of F_{ST} is between 0 and 1; F_{ST} equal to 0 means that all the subpopulations have the same allele frequencies and there is no substructure. F_{ST} equal to 1 means that the subpopulations genetically diverge, meaning that the relative allelic frequencies are different.

9.2.2.2 STRUCTURE

The STRUCTURE software, based on a Bayesian clustering approach, uses a limited set of unlinked genetic markers to assign individuals on the basis of their genotypes to populations characterized by a set of allele frequencies at each locus, while simultaneously estimating population allele frequencies. First a clustering algorithm estimates the number K of subpopulation in which the population is structured. Then, using the estimated allele frequencies the likelihood that a given genotype (for all individual and all locus) originates in a particular subpopulation is calculated using a bayesian approach by calculating the conditional probability $P\{x_{il} = j|Z, P\}$ where Z is the original population, P are the frequencies of all the subpopulation and x_i is the genotype of the individual i at the locus l. Finally, the probability $P(z_i = k)$ of each individual to belong to a particular subpopulation is computed starting from the condition that all these probabilities are equal to $1/K$, where z_i is the population from which the individual i originates. Individuals of unknown origin can be assigned to a specific population according to these likelihoods. In this way, it is possible to estimate the substructure of the original population, but it is not possible to correct for PS.

9.2.2.3 Eigensoft

The Eigensoft package (version 2.0 for Linux platform, Department of Genetics, Harvard Medical School, Boston, USA) assesses stratification by performing a PCA with the highest possible number of SNPs. PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. PCA was invented in 1901 by Karl Pearson30. PCA involves the calculation of the eigenvalue decomposition of a data covariance matrix. Eigensoft uses PCA to reduce the number of variables that describe the sample (300K SNPs scattered along the genome) in fewer dimensions that allow clustering the individuals on the basis of their genetic variance. The package contains many tool, the most important for our task are the smartpca and eigenstrat. SMARTPCA has been used to perform PCA on genotype data and to generate eigenvectors (principal components, PCs) and eigenvalues. To estimate the statistical significance of the population divergence in PC scores, analysis of variance (ANOVA) is performed among individuals divided in cases and controls and also according to the ethnic groups. Along each PC, a comparison between means and variances within subgroups (case/control and ethnic group) are computed in order to estimate the population differences. We represent the scree plot of the eigenvalues of PCA to evaluate which are the PCs that describe the largest genetic variance and to confirm the ANOVA results. A scree plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each PC. We used the software to adjust genotypes and phenotypes by variation attributable to ancestry along each PC, by computing residuals of linear regressions. Adjusted genotype is given by:

$$g_{ij}^{(\mathrm{adj})} = g_{ij} - \gamma_i a_j \qquad \gamma_i = \frac{\sum_j a_j g_{ij}}{\sum_j a_j^2}, \tag{9.3}$$

where g_{ij} is the genotype of individual j at SNP i, a_j is the ancestry (the eigenvector) of individual j along a given axis of variation and γ is the regression coefficient for ancestry predicting genotype across individuals at SNP i. Therefore, computed association with Armitage chi-square statistics between genotypes and phenotypes, both before and after PCA correction, results in two chi-square values for each SNP and relative p values. If we compare the most significantly associated SNPs in EIGENSTRAT, we observe that the results change if we correct using 3 or 9 PCs (Table 9.10), while the results using 9 PCs and 10 PCs (where λ is 1.000) are the same.

9.2.2.4 PLINK and CMH

PLINK performs a complete-linkage hierarchical clustering based on genome-wide identical-by-state (IBS) between any two subjects in order to cluster individuals into homogeneous subsets using classical multi-dimensional scaling (MDS) to visualize substructure. The CMH test is used for overall disease/gene association, controlling for clusters, where the number of clusters is selected by the user. It is a chi-squared value given by:

$$\chi^2_{\mathrm{CMH}} = \frac{\left(\sum_{i=1}^{L} a_i - \sum_{i=1}^{L} A_i\right)^2}{\sum_{i=1}^{L} V_i}, \tag{9.4}$$

where a_i are the observed count of alles, A_i are the expected count and V_i are the variance.

9.2.3 *Genomic Control*

GC 11 assumes that, in the presence of population substructure, the standard chi-squared statistics used in case–control studies is inflated by an estimated inflation factor (λ) that is proportional to the degree of stratification. Computation of λ is as described in Devlin et al. [11]:

$$\hat{\lambda} = \frac{\mathrm{median}(Y_1^2, Y_2^2, \ldots, Y_L^2)}{0.456}, \tag{9.5}$$

where Y_i^2 represents the chi-squared statistics of Armitage trend test for L unlinked markers. Then, λ value represents the median of chi-squared statistics divided by 0.456, the predicted median value of a central chi-squared distribution, A λ value above 1 indicates inflation in chi-squared statistics and, by definition, it is not allowed to be more than 1 in a homogeneous sample.

The program uses the lambda value to correct for background population differences by rescaling the chi-squared statistics of the disease–marker association test. GC corrects for stratification by adjusting association statistics at each marker with a uniform overall inflation factor.

We applied the gc.perl software (Eigensoft package) that performs GC on chi-squared statistics calculated with Eigensoft and PLINK (both before and after stratification correction). We obtained for both EIGENSTRAT and PLINK statistics the λ inflation values and chi-squared statistics after scaling by λ (for both uncorrected and corrected statistics).

9.3 Results

9.3.1 Stratification Detection

We performed case–control associations studies for schizophrenia in two different samples: the UCI sample, composed of 105 cases and 91 controls, and the CATIE-NIMH sample composed of 741 cases and 751 controls [28]. The sample sizes are different but both samples are composed of different ethnic groups. Prior to searching for SNPs associated with schizophrenia, we used PCA of EIGENSOFT package and MDS of PLINK to illustrate the genetic relatedness among individuals using the top axis of variation and to gain insight into the differences associated with ethnicity. Here, we report only PCA results, because MDS produced the same clustering. Using PCA, individuals were clustered on the basis of the inter-individual genetic variance. If we consider the sample partitioning in cases and controls, ANOVA for population differences along each principal component revealed that cases and controls in the UCI sample do not have significant differences along the PC (Table 9.2a) because they are equally distributed in the clusters (Fig. 9.1a). On the contrary, in the CATIE-NIMH sample, the ANOVA analysis revealed three major components, and in particular PC2 and PC3 reflect the major variance between cases and controls (Table 9.2a and Fig. 9.1). In the figures each point represents an individual. In the panels a and b, the colors correspond to the subdivision in cases and controls (red for cases and green for controls) both in UCI sample and CATIE-NIMH. In panel c, the colors represent the three ethnic groups in the UCI sample: red for Caucasians, blue for African Americans and green for Asians; while in the panel d, red shows

Table 9.2 p-Value of ANOVA statistics for population differences along each eigenvector (PC)

	(a) CA/CO		(b) ETHNICITY	
	UCI	CATIE	UCI	CATIE
PC1	0.303	0.0367	6.95E−17	0
PC2	0.039	8.54E−17	1.90E−18	0
PC3	0.427	5.87E−04	8.05E−06	5.30E−17
PC4	0.124	0.887	2.43E−03	0.855

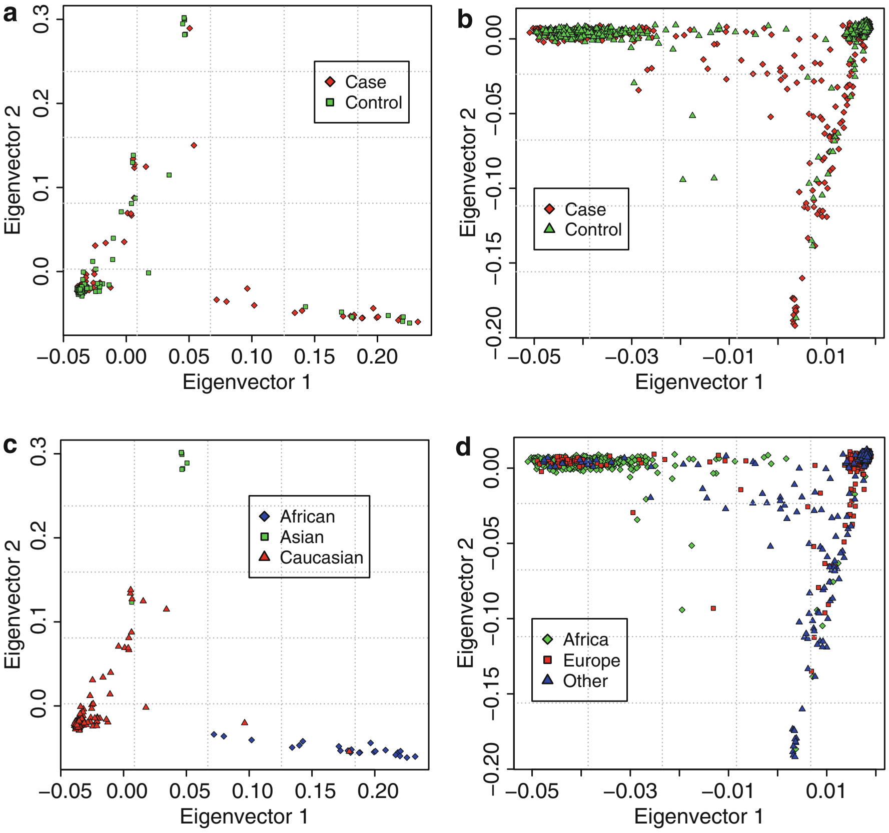


Fig. 9.1 The top two principal components (eigenvectors 1 and 2) for 107 patients with schizophrenia and 91 healthy controls of UCI sample (**a**, **c**) and for 741 cases and 741 controls of CATIE-NIMH sample (**b**, **d**)

the Europeans, green the Africans and blue the other or more than ethnicities in the CATIE-NIMH sample. A significant *p*-value makes us reject the null hypothesis of similarity between (a) cases and controls (CA/CO) and (b) ethnic groups, in both UCI and CATIE-NIMH samples.

On the other hand, if we consider the samples according to ethnicities, ANOVA statistics points to three major axis of variation in both samples (Table 9.2b). The top two axes of variation (PC1 and PC2), representing most of the genetic variance in the UCI sample ($p < 10^{-12}$), are shown in Fig. 9.1c. Interestingly, individuals clustered in accordance with their ethnic origins: in particular, PC1 provided good separation between African-Americans and Caucasians, while PC2 separated Caucasians and Asians. In the CATIE-NIMH sample, as shown in Fig. 9.1c, there is a subdivision along the PC1 between European and Africans.

We represent the scree plot of the eigenvalues of PCA to evaluate which are the PCs that describe the largest genetic variance and to confirm the ANOVA results.

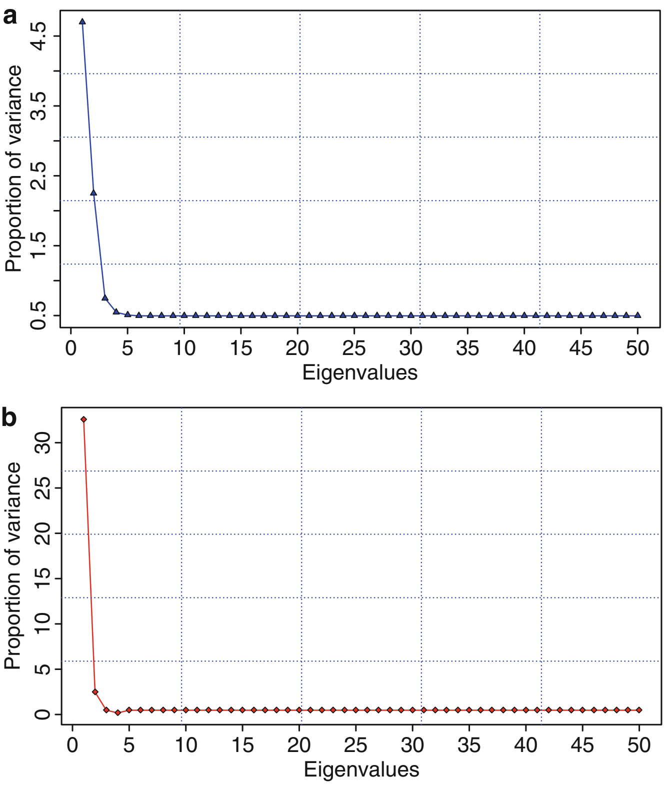


Fig. 9.2 Scree plot of eigenvalues for 50 principal components of EIGENSOFT analysis on (**a**) UCI sample and (**b**) CATIE-NIMH sample. The value of eigenvalue represent the importance of the related PC to describe the most genetic variance

A scree plot is a simple line segment plot that shows the fraction of total variance in the data as explained or represented by each PC. In our case, in both samples, the first three components describe most of the genetic variance (Fig. 9.2).

We can conclude that, in both samples, the positions of individuals in the PC space strongly correlate with the ethnic/geographic distances between populations. Then, we assessed with PCA and MDS that both samples are stratified. We confirmed the existence of a substructure with the STRUCTURE software (Fig. 9.3). In the figure, we show the analysis results of (a) the single population of Caucasians and (b) the entire sample. In the upper part, each individual is represented by a single vertical line broken into K colored segments (green=K1, red=K2, blue=K3), with lengths proportional to each of the K inferred clusters. In (b) the numbers refer to the populations: two are Asians, four the Africans and five the Caucasians. In the lower part, each individual is represented by a point. The three edges of triangle represent the K=3. In (b) the colors correspond to the prior population labels (Caucasians=green, Africans=red and Asians=blue). The estimated proportion of ancestry in each population is given by the distance to the one edge of the triangle.

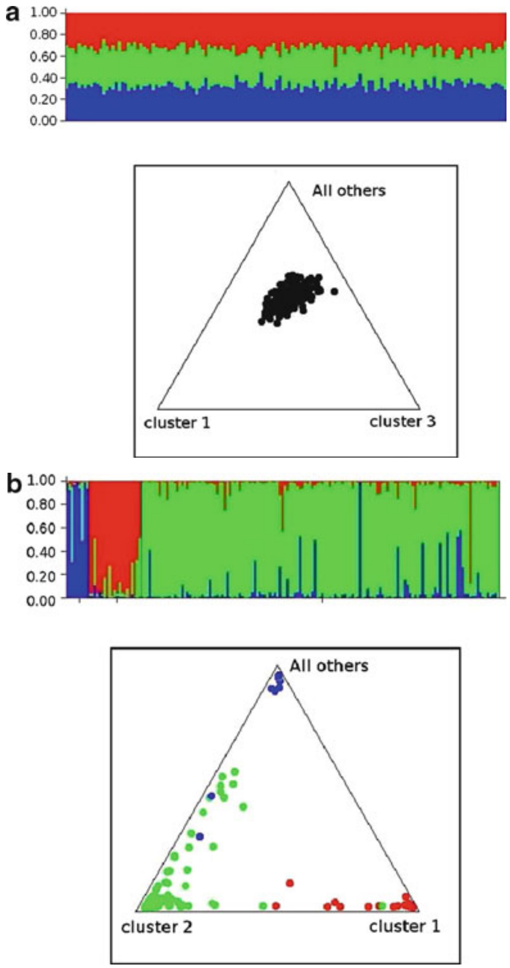


Fig. 9.3 Summary plot of estimates of Q (estimated population ancestries) in UCI sample for three clusters: (**a**) clustering obtained with a single ethnic group shows the homogeneity of each group (**b**) clustering obtained with the entire sample shows the genetic divergence between groups

With STRUCTURE, we analysed each population sample separately (for instance, in the UCI sample, Caucasians, Africans and Asians separately) as well as the entire subset, using a subset of 400 selected markers. We observed that each single population is homogeneous (Fig. 9.3a), even though it differs compared to the other populations (Fig. 9.3b): if we consider a single ethnic group, we observe that the individuals belong to a unique cluster (Fig. 9.3a), while for the entire sample the software is able to distinguish the differences between the different ethnic groups (supplementary Fig. 9.3b) as in the PCA plot (Fig. 9.1) with a limited number of well-selected SNPs describing the genetic variance between different groups.

9.3.2 Correction for Stratification

Once assessed the substructure of our sample, we performed a stratification correction using EIGENSTRAT and PLINK. Using EIGENSTRAT, the three statistically significant axes of variation (PC1, PC2, PC3) that reflect the largest variance between individuals (Table 9.2 and Fig. 9.2) were used to correct the Armitage trend chi-squared statistics for both samples. In the original CATIE paper, Sullivan et al. used used 7 PCs to correct the association results. We also performed the CMH test in PLINK controlling for the presence of clusters, which correspond to the ethnicities/subdivisions in both samples (UCI: Caucasians, Africans and Asians; CATIE-NIMH: Europeans, Africans and other/more than a single ethnicity). A threshold of $p < 10^{-4}$ was used to select the most significant associations for both EIGENSTRAT and PLINK and to create a list of top SNPs to compare. Then, we compared the resulting lists of the most significantly associated SNPs ($p < 10^{-4}$). Using EIGENSTRAT and PLINK uncorrected and corrected statistics (Table 9.3) in both datasets, we identified three different kinds of SNPs: first, there are SNPs that can be considered "true positive," because they are associated before and after correction. Then, there are SNPs that can be considered "stratified," because they are significant before correction but they are adjusted as not significant by PCA correction (false positives). Finally, there are SNPs that become significant only after the correction, while in absence of correction for PS they are hidden by stratification: these can be considered "false-negatives." If we compare the results, we observe that in the UCI sample, PLINK identifies a larger number of false positives than EIGENSTRAT, while in the CATIE-NIMH sample, in which we removed the outliers individuals, we find an opposite behavior because PLINK identifies a larger number of true positives and false negatives than EIGENSTRAT that presents a larger number of true positives. In the table, the row "COMMON" reports the number of identical SNPs found significantly associated using both methods. The values are the number of top SNPs and the percentage, in parentheses, relative to the total. True positives are those SNPs that are strongly associated with disease;

Table 9.3 Top significantly associated ($p < 10^{-4}$) SNPs with schizophrenia applying EIGENSTRAT and PLINK in UCI and CATIE-NIMH samples. In the table the following abbreviations are used: A=associated, NA=not associated, TP=true positive, FP=false positive, FN=false negative, S=stratified, NS=not stratified

		A NS TP	A S FP	NA NS FN	# TOT top significantly associated SNPs
CMH (PLINK)	UCI	12 (17.65)	44 (64.71)	12 (17.65)	68
	CATIE	1,598 (68.48)	198 (8.45)	539 (23.07)	2,332
EIGENSTRAT[a]	UCI	11 (28.21)	17 (43.59)	11 (28.21)	39
	CATIE	26 (8.49)	249 (81.37)	31 (10.13)	306
COMMON	UCI	8 (7.48)	11 (10.28)	4 (3.74)	107
	CATIE	21 (0.80)	21 (0.80)	6 (0.23)	2,638

[a] The values refer to the results after outliers removal

Table 9.4 Comparison of the number of the most significant associated SNPs ($p < 10^{-4}$) after correction with EIGENSTRAT keeping (**a**) or removing (**b**) the outlier individuals

	A NS TP	A S FP	NA NS FN
(a) With outliers	31	1,612	30
(b) Removed outliers	26	249	31

false positives are the SNPs considered stratified because they are adjusted as not significant by PCA correction; false negatives are the SNPs that became significant only after the correction, due to the presence of stratification noise that hides the putative true association.

Notably, we observed that in EIGENSTRAT, the number of false positives decreases markedly if we remove the outliers (Table 9.4).

To consider an individual as an outlier, the default value of the parameter σ of the smartpca software (i.e., the number of standard deviations which an individual must exceed along one of the top three principal components) must be equal or higher than 6.0. The values shown in the table are the number of the top SNPs. True positives are those SNPs that are strongly associated with disease; false positives are the SNPs considered stratified, because they are adjusted as not significant by PCA correction; false negatives are the SNPs that became significant only after the correction, due to the presence of stratification noise that hids the putative true association.

In the UCI sample, PLINK and EIGENSTRAT share eight SNPs truly associated with disease, 11 SNPs that are stratified and four SNPs that are false negatives (Table 9.3). In the CATIE sample, we observe the presence of 21 SNPs associated with disease, 21 false positives and six false negatives that are common between the two methods (Table 9.3). As a preliminary step of an association analysis, it is useful to focus on the overlapping true positive and false negative SNPs detected from both methods to increase the reliability/confidence of SNPs significantly associated with the disease. However, it is relevant also to compare the differences and understand why the methods give different results. We applied GC implemented in Eigensoft on the respective Armitage chi-squared statistics and the GC of PLINK on the basic allelic test chi-square (both with a 1 degree of freedom). In particular, GC corrects for stratification by adjusting association statistics at each marker with a uniform overall inflation factor by rescaling the chi-squared statistics of the disease-marker association test: in the results of the most significantly associated SNPs, we can find only true positive and false positive SNPs but not false negatives, because the GC correction provides only a rescaling of unadjusted results (Table 9.5). The row "COMMON" reports the number of identical SNPs found significantly associated using both methods. The value are the number of top SNPs and the percentage, in parentheses, relative to the total. True positives are SNPs that are strongly associated with disease; false positives are the SNPs considered stratified because they are adjusted by PCA correction; false negatives are the SNPs that became significant only after the correction, due to the presence of the stratification noise that hides the true association.

Table 9.5 Top significantly associated ($p < 10^{-4}$) SNPs with schizophrenia applying genomic control (GC) on the basic allelic chi-squared statistics (PLINK) and on the Armitage chi-squared statistics (EIGENSTRAT), in UCI and CATIE-NIMH samples

		A NS TP	A S FP	# TOT top significantly associated SNPs
Basic allelic χ^2 statistics (PLINK)	UCI	16 (29.57)	40 (71.43)	56
	CATIE	161 (8.97)	1,633 (91.03)	1, 794
Armitage χ^2 statistics (EIGENSTRAT)[a]	UCI	13 (46.43)	15 (53.57)	28
	CATIE	6 (2.18)	269 (97.82)	275
COMMON	UCI	9 (10.71)	7 (8.33)	84
	CATIE	5 (0.25)	138 (6.67)	2, 069

[a]The values refer to the results after outliers removal

Table 9.6 Top significantly associated ($p < 10^{-4}$) SNPs with schizophrenia applying genomic control (GC) on the basic allelic chi-squared statistics (PLINK) and on the Armitage chi-squared statistics (EIGENSTRAT), in UCI and CATIE-NIMH samples

	# top significant SNPs after GC correction on allelic chi-squared statistic	# top significant SNPs after CMH correction	# COMMON significantly associated SNPs
UCI	16	24	10
CATIE	161	5, 926	161

The values refer to the results after outliers removal

We observe that the GC applied on chi-squared statistics of EIGENSTRAT (Armitage chisquare statistics) and PLINK (basic allelic test chi-square) produces datasets composed of different top SNPs: there are only nine true positive and seven false positive across 84 significantly associated SNPs that are common between the two statistics from both methods. This may depend also from the specific association statistics on which GC is applied, because the statistics of Armitage test (EIGENSTRAT) and basic chi-square (PLINK) are obtained by different algorithms.

To evaluate whether the GC correction shows similar results with CMH test (Table 9.6) and EIGENSTRAT (Table 9.7), we compared the results: we observe that GC applied on the allelic chi-square shares with CMH test 10 SNPs (33.3%) in UCI sample (Table 9.6), while in the CATIE sample, all the 161 top SNPs (2.7%) obtained with the GC correction are top SNPs also using the CMH correction. On the other hand, GC and EIGENSTRAT correction on Armitage chi-square statistics present eight common SNPs (29.6%) in the UCI sample and four (6.8%) in the CATIE sample that are common between the two methods.

To assess the power of EIGENSTRAT and PLINK to correct for substructure and to verify the results, we calculated the genomic inflation factor λ on the association results for both samples (Table 9.8).

The λ value above 1 indicates inflation in chi-squared statistics and, by definition, λ is not allowed to be more than 1 in a homogeneous sample. In the UCI sample,

Table 9.7 Comparison of the number of the most significant associated SNPs after correction with GC and correction using 3 PCs in EIGENSTRAT

	# top significant SNPs after GC correction on allelic chi-squared statistic	# top significant SNPs after EIGENSTRAT correction	# COMMON significantly associated SNPs
UCI	13	22	8
CATIE	6 (102)	57 (61)	4 (3)

In the columns of EIGENSTRAT, the values refer to the results after outliers removal whereas the values in parentheses refer to the results without removal of outliers

Table 9.8 Genomic inflation factor, values, calculated on PLINK and EIGENSTRAT chi-squared association statistics in UCI (**a**) and CATIE-NIMH (**b**) sample, both before and after using 3 PCs to correct

	(a) UCI		(b) CATIE-NIMH	
	PLINK	EIGENSTRAT	PLINK	EIGENSTRAT
Before correction	1.136	1.075	1.737	1.757
After correction	1.012	1.008	1.639	1.046

both methods are able to correct for PS because the λ value is close to 1. On the contrary in the CATIE-NIMH sample, the chi-squared statistics of PLINK is inflated by substructure also after correction ($\lambda = 1.639$). This finding can explain the previous results (larger number of true positives and false negatives in Table 9.3) and shows that the CMH method in PLINK may not be powerful enough to correct for PS. This can be due to the dependence of CMH test from the user-defined number of cluster that cannot identify the presence of some hidden stratification. Indeed, if we set the number of clusters to 3 (Europe, African and other ethnicities), we are not able to find out the substructure within the group "other or more than one ethnicity." Considering the inflation factor λ, an open issue is how to choose an appropriate threshold of inflation factor to consider a sample as substructured and the resulting association inflated by stratification. To understand this, we calculated the λ value in EIGENSTRAT for UCI and CATIE-NIMH sample (Table 9.3) on the chi-squared statistics after correction using 1–10 PCs (Table 9.9).

In the UCI sample, the calculated λ value using as covariates the three major components (see ANOVA statistics in Detection stratification) has a value of 1.008 but λ decrease to 1.000 only using nine PCs as covariates. On the other hand, in the CATIE-NIMH sample, we do not reach the limit of 1 also using 10 PCs (1.033) either keeping or removing outlier individuals. If we compare the most significantly associated SNPs in EIGENSTRAT, we observe that the results change if we correct using 3 or 9 PCs (Table 9.10), while the results using 9 PCs and 10 PCs (where λ is 1.000) are the same.

Then, the problem is how much λ can deviate from 1 to consider stratification present and also if we can accept the results corrected using 3 PCs as not inflated by stratification, because we can observe that a little fluctuation of the λ value can

Table 9.9 Inflation factor (λ) values calculated in EIGENSTRAT chi-squared statistics after correction using 1 PCs (K1) to 10 PCs (K10) in the UCI sample and in the CATIE-NIMH sample. In the CATIE-NIMH sample we show the results both without or with removal of outliers

λ			
	UCI	CATIE-NIMH (+Outliers)	CATIE-NIMH (−Outliers)
K1	1.063	1.067	1.098
K2	1.01	1.067	1.036
K3	1.008	1.046	1.036
K4	1.005	1.041	1.035
K5	1.002	1.04	1.036
K6	1.001	1.038	1.034
K7	1.001	1.038	1.034
K8	1.003	1.037	1.033
K9	1	1.037	1.033
K10	1	1.037	1.033

Table 9.10 Comparison of most significantly associated SNPs in chi-squared statistics of EIGENSTRAT after correction with 3 or 9 PCs, in the UCI sample

	A NS TP	A S FP	NA NS FN
K3	11	17	5
K9	7	21	15
K3 and K9	5	11	6

cause a loss or gain of significantly associated SNPs and then to genes associated with disease (Table 9.10). In particular, higher the lambda value higher is the correction factor for chi-squared statistics. Given a fixed threshold for the significance (p-value<10E−4), the number of significantly associated SNPs decreases with an increased value of inflation factor.

9.4 Conclusions

Case–control studies are hampered by PS that can occur in populations and can lead to significant associations being detected at loci that have nothing to do with disease. There is currently no consensus about the effectiveness of a specific strategy that allows to correct for PS. Our goal, studying different populations and complex diseases, is to find an efficient strategy that allows us to correct our ethnically mixed samples and thus to avoid false-positive genotype–phenotype associations. Our current study shows an application of different methods to measure stratification when genotyping thousands of genetic markers in two American case–control samples composed of different ethnic groups. The study, involving PCA and MDS identification of different subpopulations in our samples, provided additional insight into the substructure of American populations and differences among various ethnic groups that may impact our understanding of the genetics of complex diseases. We also emphasize the importance of controlling substructure in the ascertainment of putative associations between genes and disease. Notably, without the correction for substructure in our study, some SNPs would have appeared as strong candidates for schizophrenia even though the large differences in allele frequency for these SNPs were largely due to differences in allele frequency among different population subgroups. The stratification correction allows to correct for false-positive associations, but it also may help in rescuing potential false-negative associations as shown in Tables 9.3 and 9.8. We also noted that the effects of PS increase with increasing sample size: as we saw in the simulation with EIGENSTRAT (Table 9.3), the number of SNPs that are stratified and then corrected (false positives) is bigger in the CATIE-NIMH than in the UCI sample.

A general assumption is that the λ value of GC can be used to verify the power of other stratification corrections as PCA or CMH methods: however, we observed that the choice of the most appropriate threshold for λ is very difficult, and even small fluctuations of λ can cause the gain or loss of different associated SNPs, and then the results can change dramatically. Notably, only the PCA analysis in EIGENSTRAT is able to identify and correct the presence of some hidden stratification, while CMH test in PLINK may not be powerful enough to correct for them because it is dependent from the user-defined number of clusters which do not consider some unknown hidden stratification.

We can conclude that:

- The main limitation of F_{ST} is that it strongly depends on the number of SNPs and it assesses only the presence of PS without correction.
- The major limitations of STRUCTURE are the intensive computational cost on large datasets if applied to genome-wide datasets, and the sensitivity to the number of clusters defined by users before analysis. However, it is based on a rigorous Bayesian statistics.
- The limitation of CMH test is the dependence to the number of cluster. It cannot identify and correct the presence of hidden stratification because the subgroups are user-defined and not calculated by the software. However, it is applicable for large genome wide dataset.
- The limitation of GC is that the correction (lambda) is uniform for all SNPs.
- The main advantage of eigensoft is the correction of both genotypes at each SNP (not uniform) and phenotypes, but it strongly depends on the used-defined number of principal component along which the correction is performed.

Then we decided to use more than one, thus adopting the following procedure, (shown also in Fig. 9.4):

1. Identification of population substructure using PCA in EIGENSTRAT or MDS in PLINK, which allow clustering the individuals on the basis of genetic variance.
2. Verification of the results in STRUCTURE using the number of cluster identified with PCA or MDS.
3. Correction of association study results with EIGENSTRAT, once identified the statistically significant principal components that describes population divergence. Then, having also performed CMH test for overall disease–gene association, controlling for clusters, the results are compared.

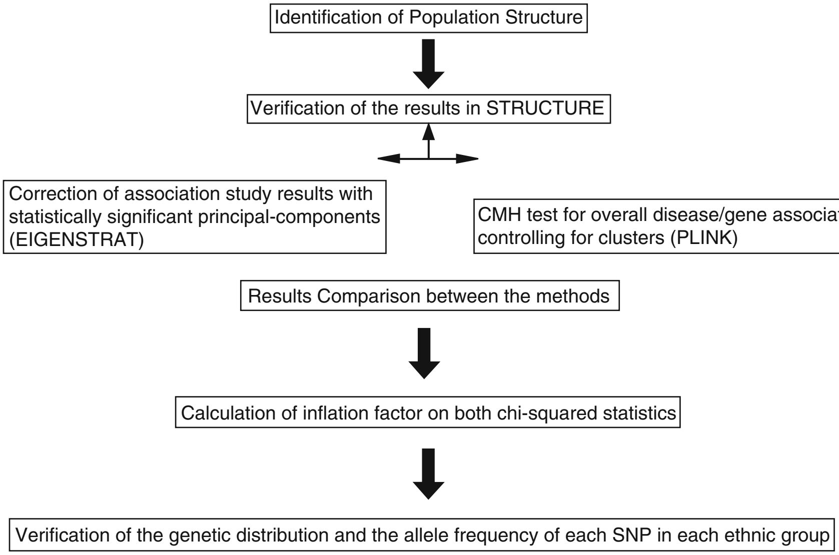

Identification of Population Structure
Verification of the results in STRUCTURE
Correction of association study results with statistically significant principal-components (EIGENSTRAT)
CMH test for overall disease/gene association, controlling for clusters (PLINK)
Results Comparison between the methods
Calculation of inflation factor on both chi-squared statistics
Verification of the genetic distribution and the allele frequency of each SNP in each ethnic group

Fig. 9.4 Workflow of PS analysis

4. The sets of significantly associated SNPs, obtained before and after correction, are compared in order to identify the SNPs that are truly associated with disease, the "stratified SNPs" related to the ethnicity and also the false negative significant SNPs. It is useful, as preliminary step, to focus on the overlapping true positive and false negative SNPs detected from both methods to increase the confidence of SNPs really relevant for the disease. Later, it is relevant to compare and understand the differences between the methods.
5. Calculation of inflation factor on both association statistics
6. Verification, for each significant SNPs, of the genetic distribution and the allele frequency in each ethnic group in order to filter out results with anomalies (for example, SNPs with not represented alleles).

Acknowledgements This work was supported by FIRB Italia-Israele [RBIN04SWHR], a fellowship of the Doctorate School of Molecular Medicine, University of Milan, POCEMON [FP7-ICT-2007-216088], HYPERGENES [HEALTH-F4-2007-201550], InGenious HyperCare [LSHM-CT-2006-037093], by the Israel Science Foundation [Israel Academy of Sciences, Grant #348/09], by the Enabling Grids for E-sciencE (INFSO-RI-222667), CNR-BIOINFORMATICS, ITALBIONET, Italian-Canada FIRB-MUR Projects.

References

1. Cardon LR, Bell JI: Association study designs for complex diseases. Nat Rev Genet 2(2): 91–99 (2001)
2. Zondervan KT, Cardon LR: Designing candidate gene and genome-wide case-control association studies. Nat Protoc 2(10): 2492–2501 (2007)

3. Ziegler A, Konig IR, Thompson JR: Biostatistical aspects of genome-wide association studies. Biom J 50(1): 8–28 (2008)
4. Potkin SG, Turner JA, Guffanti G, Lakatos A, Torri F, Keator DB, Macciardi F: Genome-wide strategies for discovering genetic influences on cognition and cognitive disorders: methodological considerations. Cogn Neuropsychiatry 14(4): 391–418 (2009)
5. Barnholtz-Sloan JS, McEvoy B, Shriver MD, Rebbeck TR: Ancestry estimation and correction for population stratification in molecular epidemiologic association studies. Cancer Epidemiol Biomarkers Prev 17(3): 471–477 (2008)
6. Freedman ML, Reich D, Penney KL et al: Assessing the impact of population stratification on genetic association studies. Nat Genet 36: 388–393 (2004)
7. Cardon LR, Palmer LJ: Population stratification and spurious allelic association. Lancet 361: 598–604 (2003)
8. Weir BS, Cockerham CC: Estimating F-statistics for the analysis of population structure. Evolution 38: 1358–1370 (1984)
9. Pritchard JK, Stephens M, Donnelly P: Inference of population structure using multilocus genotype data. Genetics 155: 945–959 (2000)
10. Li JZ, Absher DM, Tang H, Southwick AM, Casto AM, Ramachandran S, Cann HM, Barsh GS, Feldman M, Cavalli-Sforza LL, Myers RM: Worldwide human relationships inferred from genome-wide patterns of variation. Science 319(5866): 1100–1104 (2008)
11. Devlin B, Bacanu B, Roeder K: Genomic control in the extreme. Nat Genet 36: 1129–1130 (2004)
12. Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA: Principal components analysis corrects for stratification in genome-wide association studies. Nat Genet 38: 904–909 (2006)
13. Purcell S, Neale B, Todd-Brown K et al: PLINK: a toolset for whole-genome association and population-based linkage analysis. AJHG 2007 81: 559–575 (2007)
14. Bauchet M, McEvoy B, Pearson LN, Quillen EE, Sarkisian T, Hovhannesyan K, Deka R, Bradley DG, Shriver MD: Measuring European population stratification with microarray genotype data. Am J Hum Genet 80(5): 948–956 (2007); Epub Mar 22 2007
15. Tian C, Plenge RM, Ransom M, Lee A, Villoslada P, Selmi C, Klareskog L, Pulver AE, Qi L, Gregersen PK, Seldin MF: Analysis and application of European genetic substructure using 300 K SNP information. PLoS Genet 4(1): e4 (2008)
16. Price AL, Butler J, Patterson N et al: Discerning the ancestry of European Americans in genetic association studies. PLoS Genet 2008 4(1): e236 (2007); Epub Nov 19 2007
17. Patterson N, Price AL, Reich D: Population structure and eigenanalysis. PLoS Genet 2: 2074–2093 (2006)
18. Novembre J, Stephens M: Interpreting principal component analyses of spatial population genetic variation. Nat Genet 40(5): 646–649 (2008); Epub Apr 20 2008
19. Yu K, Wang Z, Li Q, Wacholder S, Hunter DJ, Hoover RN, Chanock S, Thomas G: Population substructure and control selection in genome-wide association studies. PLoS ONE 3(7): e2551 (2008)
20. Wellcome Trust Case Control Consortium.: Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature 447(7145): 661–678 (2007)
21. Yeager M, Orr N, Hayes RB et al: Genome-wide association study of prostate cancer identifies a second risk locus at 8q24. Nat Genet 39(5): 645–649 (2007); Epub Apr 1 2007
22. Hunter DJ, Kraft P, Jacobs KB et al: A genome-wide association study identifies alleles in FGFR2 associated with risk of sporadic postmenopausal breast cancer. Nat Genet 39(7): 870–874 (2007); Epub May 27 2007
23. Epstein MP, Allen AS, Satten GA: A simple and improved correction for population stratification in case-control studies. Am J Hum Genet 80(5): 921–930 (2007); Epub Mar 29 2007
24. Serre D, Montpetit A, Par G, Engert JC, Yusuf S, Keavney B, Hudson TJ, Anand S: Correction of population stratification in large multi-ethnic association studies. PLoS ONE 3(1): e1382 (2008)
25. Li Q, Yu K: Improved correction for population stratification in genome-wide association studies by identifying hidden population structures. Genet Epidemiol 32(3): 215–226 (2008)

26. Seldin MF, Price AL: Application of ancestry informative markers to association studies in European Americans. PLoS Genet 4(1): e5 (2008)
27. Mao X, Bigham AW, Mei R, Gutierrez G, Weiss KM, Brutsaert TD, Leon-Velarde F, Moore LG, Vargas E, McKeigue PM, Shriver MD, Parra EJ: A genomewide admixture mapping panel for Hispanic/Latino populations. Am J Hum Genet 80(6): 1171–1178 (2007); Epub Apr 20 2007
28. Sullivan PF, Lin D, Tzeng JY et al: Genomewide association for schizophrenia in the CATIE study: results of stage 1. Mol Psychiatry 13: 570–584 (2008)
29. Steemers FJ, Gunderson KL: Pharmacogenomics 6: 777–778 (2005)
30. Fan J-B, Chee MS, Gunderson KL: Highly parallel genomic assays. Nature Publishing Group 7: 632–644 (2006)

Chapter 10
Predicting and Measuring the Sequence Distribution of Addition Polymers

Maurizio S. Montaudo

Abstract The sequence distribution of poly(styrene), poly(methyl methacrylate) and other addition polymers can be predicted, starting from the knowledge of polymerization reaction conditions. In many cases, the sequence distribution will be Markovian (of the first or second order), but in other cases, it cannot be described by Markovian statistics. Three examples of sequences falling in the latter class are discussed. All types of copolymers are considered: AB copolymers, ABC copolymers, ABCD copolymers. As reaction time increases, polymerization dynamics becomes less trivial. Additional parameters are required to describe how copolymer sequence varies as the reaction yield (or the reaction time) increases. Nevertheless, reaction products are conceptually simple points, and it is possible to follow their changes by drawing their trajectories in a multidimensional phase space. The task of measuring the sequence distribution is seldom trivial. Many examples of polymer sequencing using NMR spectroscopy have been collected and discussed by Randall. Mass spectrometry is also used. Often sequence distribution information must be extracted from experimental data. Flexible empirical models have been developed for this aim. Mixtures of two bernoullian chains and mixtures of two markovian chains are used. The *pertubed markovian* model features ε, a perturbation factor. Some experimental methods attempt to measure polymer sequence by partial degradation, i.e., by reducing the length of the chains until a mixture of tetramers, pentamers and hexamers is obtained. This procedure yields a new copolymer, with a new sequence distribution. The sequence of the undegraded polymer must be reconstructed from the knowledge of the sequence of the partially degraded one.

M.S. Montaudo (✉)
Institute of Chemistry and Technology of Polymers, CNR,
Via Paolo Gaifami, 18-95126 Catania, Italy
e-mail: mmontaudo@unict.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_10,

10.1 Introduction

Polymers are made of identical units that repeat themselves along the macromolecular chain. In branched polymers, the chain can be Y-shaped, X-shaped, etc. In linear polymers the chain is straight, since linear polymers are characterized by the absence of branching points. Polymers are obtained reacting monomers. Copolymers (see below) are obtained reacting co-monomers. Linear polymers made of more than one million repeat units have been synthetized. For instance poly(siloxane) and poly(ethylene). In oligomers, the chain is short (by definition). The dimer contains two repeat units, whereas the trimer, tetramer and the pentamer contain three four and five repeat units, respectively. Textbooks in polymer science classify polymerization reactions in two groups, namely chain-growth polymerization and step-growth polymerization [1–3]. In chain-growth polymerizations (also referred to as addition polymerizations), the chain grows in a well-defined manner by adding one monomer at a time. The monomer always contains a C=C bond (triple bonds are not considered here). In step-growth polymerization, a small molecule (e.g., water, methanol or chloridic acid) is often eliminated. Furthermore, the chain grows by adding monomers, but also dimers, trimers, and tetramers (for instance, a trimer can react with a tetramer to form an heptamer). Figure 10.1 reports the structures of three common polymers, namely poly(styrene), poly(methylmethacrylate) and poly(ethyleneoxide). Binary copolymers are polymers constituted by only two types of units, denoted by A and B, repeating themselves along the macromolecular chain. In exactly alternating copolymers, the sequence is ABABABABABABABABA. In random copolymers, A and B units are found at random along the chain. In block copolymers, long AAAAAAA and BBBBBB blocks are present. The group of block copolymers comprises diblock copolymers AAAAAAABBBBBBB, triblock copolymers AAAAAAABBBBBBBAAAAAA and more complex copolymers. A *tri-penta-tapered* copolymer is a copolymer which starts with a block of at least five consecutive A units and ends with a block of at least five consecutive B units and which possesses at least three AB heterodiads. It is obvious that triblock copolymers and diblock copolymers are not tri-penta-tapered copolymer because they do not fit in the definition. An interesting tri-penta-tapered copolymer is SS/DD defined [4] as *a taperered block copolymer that tapers from a polystyrene block to a styrene-diene random copolymer to a polydiene block*. Tri-penta-tapered copolymer belong to a large class of copolymers called gradient copolymers. Moving along

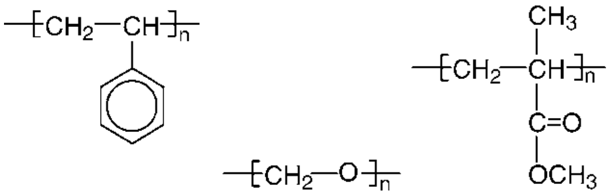


Fig. 10.1 Structures of three common polymers, namely poly(styrene), poly(methylmethacrylate) and poly(ethyleneoxide)

the chain, the relative abundance of A and B units changes. For instance, a gradient copolymer with units of styrene (ST) and butadiene (BU) was obtained [5]. The copolymer ends with a block of at least five consecutive ST units but it does not start with five consecutive BU units, and, thus, it does not belong to the set of tri-penta-tapered copolymers. In ABC and ABCD copolymers, three and four different repeat units are found along the macromolecular chain, respectively. During the years, various polymerization reactions have been developed, which yield different sequences. Theories have been put forward to model polymerization reactions. It will be shown that each theory predicts a different sequence. Thereafter, we will discuss experimental techniques for measuring the sequence of addition polymers. We will concentrate on spectroscopic methods, namely fourier-transform infrared spectroscopy (FTIR), nuclear magnetic resonance (NMR) and mass spectrometry (MS). Each measurement has advantages and disadvantages. These will be discussed too.

10.2 Sequence Prediction

A polymerization reaction is a chemical reaction in which reactants are transformed in reaction products. The reactants are called the feed. The feed contains monomers at a concentration z_{tot}. In binary copolymers, the feed contains A and B at a concentration z_{A} and z_{B}, and thus the sum of the molar ratio of A and B units in the feed $f_{\text{A}} + f_{\text{B}}$ equals 1. In a similar manner, the sum of the molar ratio of A and B units in the reaction products (i.e., in copolymer chains) $c_{\text{A}} + c_{\text{B}}$ equals one, indeed. For chemical industries that produce and sell copolymers, optimization is important. The temperature and the pressure inside the reaction vessel are continuously monitored and varied during the course of the reaction. Often the reaction vessel is equipped with a mechanical stirrer that ensures good spatial homogeneity for both the reactants and the reaction products. The reaction vessel is opened at time $t = t_{\text{add}}$, and some new reactants are added. The sequence of addition polymers can be predicted, starting from the knowledge of all these reaction conditions. Papers on this vast topic usually appear chemical engineering journals [6–8], but some papers are published on macromolecular journals [9, 10]. Unfortunately, the predictions are invariably numerically oriented.

On the other hand, some polymerization reactions can be described by simple models, so simple that an algebraic solution can be derived. In the following, we shall describe them and we will derive an analytical prediction for the sequence of the reaction products. In order to cast the prediction in a homogeneous form, the sequence must be carefully normalized. Let us consider a medium-sized polymeric sample (e.g., 100 g) made of medium-sized chains and focus on the probability of occurrence of a particular sequence XXXX. The weight (in grams or milligrams), W_{XXXX}, of an oligomer with that specific sequence can be safely factored in two parts:

$$W_{\text{XXXX}} = \text{MMD}(s) I_{\text{XXXX}}, \tag{10.1}$$

where s is the size of the sequence, MMD(s) is the distribution of molar masses and I_{XXXX} is the normalized probability. MMD(s) has the same dimensions of W_{XXXX}, whereas I_{XXXX} is dimensionless. In AB copolymers, X can take two values X=A and X=B. In ABC copolymers, X can take three values X=A, X=B and X=C. A sequence made of three repeat units is called triad. Tetrads, pentads, hexads and heptads are sequences made of four, five, six and seven repeat units, respectively. W_{XXXX} and I_{XXXX} are square matrices for dyads in binary copolymers, for triads in ABC copolymers, for tetrads in ABCD copolymers, for L-ads when the length of the sequence is L. In the other cases, W_{XXXX} and I_{XXXX} are rectangular matrices.

The MMD can be measured by chromatography, by MS or by other means [1–3]. MMD averages are very important. The number-average molar mass, $\bar{\prod}_n$, is given by:

$$\bar{\prod}_n = \left(\sum \mathbf{m}_i N_i\right) \Big/ \left(\sum N_i\right) \tag{10.2}$$

In a similar manner, the weight-average molar mass, $\bar{\prod}_w$, is given by:

$$\bar{\prod}_w = \left(\sum (\mathbf{m}_i)^2 N_i\right) \Big/ \left(\sum \mathbf{m}_i N_i\right), \tag{10.3}$$

where all summations span over all masses (from one to infinity). The above equations can be used to measure the MMD of a polymer sample by simply measuring the abundance N_i of each macromolecular chain with mass $\mathbf{m}_i$. The most important MMD is the Schulz–Flory MMD, which is a decreasing exponential. The ratio $\bar{\prod}_w/\bar{\prod}_n$ takes the value 2. The Schulz–Zimm MMD function is given by the product of a power-law and a decreasing exponential:

$$\mathrm{MMD}(s) = a_{\mathrm{nofa}}(s)^{\alpha}\exp(-s/\theta), \tag{10.4}$$

where α and θ are two adjustable parameters and a_{nofa} is a suitable normalization factor. The MMD averages, computed using the definition, turn out to be:

$$\bar{\prod}_n = \mu(\alpha + 1)/\theta \tag{10.5}$$

$$\bar{\prod}_w = \mu(\alpha + 2)/\theta, \tag{10.6}$$

where μ is the mass of the repeat unit. In the case of poly(styrene), poly(methyl methacrylate) and poly(ethyleneoxide) (see figure), μ is 108, 100 and 44 g mol^{-1}, respectively

10.3 Free-Radical Copolymerization

Among chain-growth polymerization reactions (also referred to as addition polymerizations), free-radical copolymerization is the most common [1–3]. Figures 10.2 and 10.3 report in a concise manner the chemical reactions that occur in free-radical

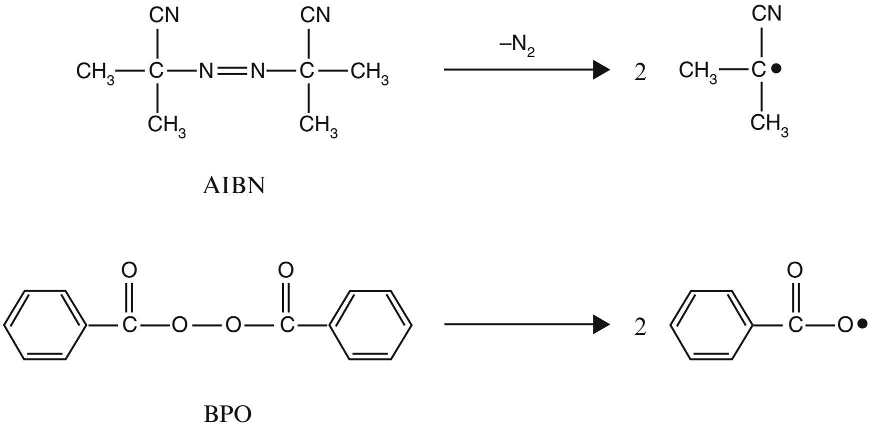


Fig. 10.2 Structures of two common initiators, namely 2,2′-azobis(isobutyronitrile) (AIBN for brief) and dibenzoyl peroxide (BPO for brief) along with the chemical reactions which occur during their thermal scission. It can be seen that each initiator molecule produces two radicals

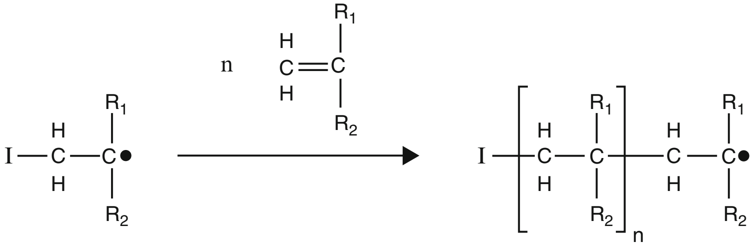


Fig. 10.3 The chemical reactions which occur in free-radical copolymerization process. The two monomers are ethylene derivatives in which the first carbon atom does not have substituents

copolymerization process. The two monomers are ethylene derivatives in which the first carbon atom does not have substituents. The second carbon atom has one or two substituents, R1 and R2. For styrene, R1 = H and R2 = phenyl ring. On the other hand, for methylmethacrylate, R1 = methyl and R2 = methacrylate. An initiator, $I^{\bullet}$, is present in the mixture. Similar to a catalysts, it starts the reaction. However, the initiator acts in a different manner since it takes part in the reaction. Figure 10.2 reports the structures of two common initiators, namely 2,2′-azobis(isobutyronitrile) (AIBN for brief) and dibenzoyl peroxide (BPO for brief). It also reports the chemical reactions that occur during their thermal scission. It can be seen that each initiator molecule produces two radicals. When photo-scission is required, other initiators are used, e.g., 2,2′-azobis(cyclohexane-l-carbonitrile). As the reaction goes on, the monomer is transformed into polymer. Figure 10.3 also reports a scheme of the polymerization reaction. In the initiation process, the initiator attacks one of the two monomers to form a initiator–monomer entity (see figure). In the propagation process, the latter reacts *n* times with a monomer to yield a growing free-radical chain made of n comonomers (see figure). The chemical reaction that ends the polymerization process is called *termination*. The reaction can be transfer to monomer, transfer to solvent, single-chain termination or two-chain termination. The latter is also called *coupling* for the following reason: the reactive head of a growing radical chain of size s_1 attacks the reactive head of a growing radical chain of size s_2 to produce a stable chain of size $s_1 + s_2$. In addition polymerization, the monomer and the repeat unit are almost identical (the difference is in the type of bond between the two carbon atoms), and therefore the former and the latter will be denoted by M and $\underline{M}$, respectively. An important quantity is Ψ, the monomer-to-polymer conversion also referred to as the polymerization yield or simply the yield. When Ψ is very high (close to 100), the molar ratio of A units in the chain, c_{A}, must necessarily reflect the molar ratio in the monomer feed, f_{A} (monomers cannot disappear). On the other hand, in 1940–1950, researchers noted that if one stops the reaction at low conversion, c_{A} in general differs from f_{A}. This implies that the propagation process is preferential and that the rates at which monomer A and monomer B are included in the growing chain are different. The relationship between f_{A} and c_{A} is called the *instantaneous composition equation*.

$$c_{\mathrm{A}} \neq f_{\mathrm{A}} \tag{10.7}$$

The propagation process is preferential and this implies that f_{A} is a function of t (the reaction time). Initially f_{A} takes the value $f_{\mathrm{A}}(0)$. At intermediate conversions, it takes the value $f_{\mathrm{A}}(t)$. Different theories have been proposed to model the free-radical copolymerization and they all predict different copolymer sequences. Many theories deal with free-radical copolymerization. A review by Kuchanov with more than 340 references on the quantitative theory of free-radical copolymerization [11] appeared in 1992. We will now discuss the principal models.

10.4 The Terminal Model

In the terminal model (sometimes referred to as the ultimate model), the rate constants of monomer addition to the growing copolymer radical depends exclusively on the monomer to be added and on the type of its terminal unit. Thus, there are four different propagation reactions

$$\sim\sim \underline{M}_1 + M_1 \overset{k_{11}}{\rightarrow} \sim\sim \underline{M}_1\underline{M}_1 \tag{10.8}$$

$$\sim\sim \underline{M}_1 + M_2 \overset{k_{12}}{\rightarrow} \sim\sim \underline{M}_1\underline{M}_2 \tag{10.9}$$

$$\sim\sim \underline{M}_2 + M_1 \overset{k_{21}}{\rightarrow} \sim\sim \underline{M}_2\underline{M}_1 \tag{10.10}$$

$$\sim\sim \underline{M}_2 + M_2 \overset{k_{22}}{\rightarrow} \sim\sim \underline{M}_2\underline{M}_2, \tag{10.11}$$

which define four different rate constants of propagation $k_{11}, k_{12}, k_{21}, k_{22}$. It is intuitive that the sequence of the copolymer resulting from reactions depicted in (10.11) is a first-order Markov chain with an associated 2 × 2 **P**-matrix and that the four **P**-matrix elements are

$$P_{\mathrm{AA}} = k_{11} f_{\mathrm{A}} / g_{2a} \;\; P_{\mathrm{AB}} = k_{12} f_{\mathrm{B}} / g_{2a} \tag{10.12}$$

$$P_{\mathrm{BA}} = k_{21} f_{\mathrm{A}} / g_{2b} \;\; P_{\mathrm{BB}} = k_{22} f_{\mathrm{B}} / g_{2b}, \tag{10.13}$$

where $g_{2a} = k_{11} f_{\mathrm{A}} + k_{12} f_{\mathrm{B}}$ and $g_{2b} = k_{21} f_{\mathrm{A}} + k_{22} f_{\mathrm{B}}$. In order to proof this intuitive result in a formal manner, it is necessary to introduce definitions for the statistical process and to demonstrate that it converges towards a stationary vector. The stationary sequence possesses an associated composition:

$$c_{\mathrm{A}} / c_{\mathrm{B}} = P_{\mathrm{AB}} / P_{\mathrm{BA}} \tag{10.14}$$

The above equation does not lend itself to immediate comparison with the equation $c_{\mathrm{A}} \neq f_{\mathrm{A}}$ (see (10.7)). Fortunately, it can be recast in an equivalent (more explicit) form:

$$c_{\mathrm{A}} = \frac{r_1 f_{\mathrm{A}}(t)^2 + f_{\mathrm{A}}(t) f_{\mathrm{B}}(t)}{r_1 f_{\mathrm{A}}(t)^2 + r_2 f_{\mathrm{B}}(t)^2 + 2 f_{\mathrm{A}}(t) f_{\mathrm{B}}(t)}, \tag{10.15}$$

where f_{A} is now time-dependent (the sum $f_{\mathrm{A}}(t) + f_{\mathrm{B}}(t)$ always equals one) and where r_1 and r_2 are the so-called *reactivity ratios* [1–3]. Equation (10.15) is the well-known *instantaneous composition equation for the terminal model.* The terminal model is extremely popular. Equations have been used in thousands of reports. Brar and coworkers used the terminal model in a systematic manner for copolymer sequencing by NMR and published more than 50 reports. Among others, reports on three copolymers containing N-vinylcarbazole units, on three copolymers containing trans-4-acryloyloxyazobenzene units, on acrylonitrile-hexyl methacrylate [12], on vinyl acetate-glycidyl methacrylate [13], on acrylonitrile-methyl acrylate [14], on acrylic acid – vinylacetate [15] on hydroxyethyl methacrylate – methyl acrylate [16].

10.5 The Bivariate Distribution

Let $W(m,n)$ denote the weight of the copolymer chains of the type $A_m B_n$ (which are made of m repeat units of type A and n repeat units of type B). The overall mass of the chain is $g_{\mathrm{all}} = m\mu_{\mathrm{A}} + n\mu_{\mathrm{B}}$, where μ_{A} and μ_{B} are the molar masses of the two repeat units. One usually assumes that all copolymer molecules produced instantaneously have the same composition. However, this is not true. Since the size $s = m + n$ of a copolymer is finite, the compositions cannot be all identical. In order to quantify this effect, one can introduce y, the deviation from the average composition, defined as $y = c_{\mathrm{A}} - \frac{m}{s}$. In order not to loose adherence with experimental measurements, it is mandatory to transform $W(m,n)$ into $W_s(y)$, the weight (in grams or milligrams) of the polymeric chains which possess a size in the range

$[s, s+\mathrm{d}s]$ and a composition deviation in the range $[y, y+\mathrm{d}y]$. $W_s(y)$ is the *bivariate distribution of chain sizes and compositions*. Integrating over sizes the bivariate distribution, one obtains CODIHI, the compositional distribution histogram.

$$\mathrm{CODIHI} = \int W_s(y)\mathrm{d}s' \tag{10.16}$$

The term histogram has historical roots. In fact, the first measurements were extremely time-consuming, and the researchers were forced to use compositional ranges of 0.05 or even 0.10. When the CODIHI is not bimodal, the following approximation holds:

$$\mathrm{FWHM} \propto \sigma, \tag{10.17}$$

where FWHM is the full width at half maximum and σ^2 is the variance of the CODIHI. A difficulty often arises. Some experimental apparata measure molar fractions and not weight fractions. There are two solutions to this problem. One can still use the definition in (10.16), along with the relationship $W(m,n) = g_{\mathrm{all}} I(m,n) \sim sI(m,n)$, which relates molar fractions and weight fractions. As an alternative, one may introduce the copolymer composition distribution (CCD) [17]:

$$\mathrm{CCD} = \int s^{-1} W_s(y)\mathrm{d}s' \tag{10.18}$$

The two solutions seem totally different. However, taking the first derivatives of (10.16) and (10.18), one finds the relationship which relates molar fractions and weight fractions. This implies that they are equivalent.

In the terminal model, $W(m,n)$ is given by the product of s **P**-matrix elements:

$$W(m,n) = \mathrm{MMD}(s) \sum a_{00} (P_{\mathrm{AA}})^{n1} (P_{\mathrm{AB}})^{n2} (P_{\mathrm{BA}})^{n3} (P_{\mathrm{BB}})^{n4}, \tag{10.19}$$

where the summation spans over $n1, n2, n3, n4$ and a_{00} is an integer number. Grouping together all sequences with the same values of n1, n2, n3, n4, approximating summations with integrals, changing the integration limits (replacing them with $-\infty$ and $+\infty$), approximating factorials using Stirling's formula, we obtain the well-known *instantaneous bivariate distribution for the terminal model* [18, 19]:

$$W_s(y) = \mathrm{MMD}(s) \exp\left(-\frac{y^2}{2\sigma^2}\right), \tag{10.20}$$

where σ^2 is defined as

$$\sigma^2 = \frac{c_{\mathrm{A}} c_{\mathrm{B}}}{s} \sqrt{(1 - 4c_{\mathrm{A}} c_{\mathrm{B}}) \left(1 - \frac{P_{\mathrm{AA}} P_{\mathrm{BB}}}{P_{\mathrm{AB}} P_{\mathrm{BA}}}\right)} \tag{10.21}$$

This is a very general result, valid for all Markovian chains $P_{\mathrm{AA}}, P_{\mathrm{AB}}, P_{\mathrm{BA}}, P_{\mathrm{BB}}$ which are not infinite. The result can be stated saying that in markovian chains, finite-size effects produce chains of the type $A_m B_n$ and the instantaneous bivariate distribution is a Gaussian curve centered at c_{A}. For dimers, trimers and tetramers, the bivariate distribution is quite broad [20]. However, the variance changes with chain size in a very strong manner (it scales as s^{-1}, see (10.21)). In the original derivation (by Stockmayer), this result was hidden. In fact, an equivalent expression for σ^2 was used in which $P_{\mathrm{AA}} P_{\mathrm{AB}} P_{\mathrm{BA}} P_{\mathrm{BB}}$ do not appear.

10.6 The Penultimate Model

In the penultimate model, the rate constants of monomer addition to the growing copolymer radical depends on its terminal unit, its penultimate unit and the monomer to be added [11]. Thus, there are eight different propagation reactions which define eight different rate constants of propagation k_{pqr}..., $p, q, r = 1, 2$. The sequence is much less intuitive than the previous one. In fact, it is described by a second-order Markov chain in which the states S1, S2, S3, S4, are *pairs* of monomer units. The chain has an associated 4 × 4 **Q**-matrix. Eight **Q**-matrix elements are zero, whereas other the eight **Q**-matrix elements are nonzero. In this case, there are four reactivity ratios r_{11}, r_{12}, r_{21} and r_{22}, defined as:

$$r_{11} = \frac{k_{111}}{k_{112}}, r_{21} = \frac{k_{211}}{k_{212}}, r_{22} = \frac{k_{222}}{k_{221}}, r_{12} = \frac{k_{122}}{k_{121}} \tag{10.22}$$

the relation between c_{A} and f_{A} for this model is

$$c_{\mathrm{A}} = \Phi_{\mathrm{penu}}(f_{\mathrm{A}}, r_{11}, r_{12}, r_{21}, r_{22}), \tag{10.23}$$

where Φ_{penu} denotes a relatively simple dependence. Penlidis and coworkers [21] investigated the role of impurities in free-radical copolymerization and they noted that Φ_{penu} can be cast in a compact form, structurally similar to (10.15) introducing two new variables, r_6 and r_7, defined as:

$$r_6 = \frac{r_{21}(f_{\mathrm{A}} r_{11} + f_{\mathrm{B}})}{f_{\mathrm{A}} r_{21} + f_{\mathrm{B}}}, r_7 = \frac{r_{12}(f_{\mathrm{B}} r_{22} + f_{\mathrm{A}})}{f_{\mathrm{B}} r_{12} + f_{\mathrm{A}}} \tag{10.24}$$

The penultimate model is slightly less popular than the terminal model, due to its higher complexity. Losio et al. [22] used it for NMR sequencing of copolymers with units of ethylene and 4-methyl-1-pentene, whereas Tritto et al. [23] and Yamada et al. [24] used it for NMR sequencing of ethylene-norbornene and of vinylacetate-vinylpivalate copolymers, respectively.

10.7 Sequence Descriptors

The only way to define a sequence distribution in a unique manner is to give the abundances for all sequences. Since the latter are infinite, one must give a model (i.e., the "building up" equations which allow to derive abundances for long sequences starting from the abundances for short sequences) and the parameters which define the model. The sequence distribution for a binary copolymer can be defined in an approximately unique manner giving the abundances of all heptads (which are 128). This is a very good approximation, since two sequences with the same heptads are identical for all purposes, but it is unpractical. There is a need to summarize, to condense [25, 26]. A scientific debate started in the early 1960s. The question was: which quantity describes the sequence best? Two papers appeared indeed very early, much earlier than others. Harwood and coworkers proposed the *run-number* [27], whereas Ring proposed a *sequence heterogeneity* [28]. During the following 40 years, many other sequence descriptors were proposed. The debate is doomed not to settle easily. The number-average lengths $< n_{\mathrm{A}} >$ and $< n_{\mathrm{B}} >$ measure the abundance of long AAAA and BBBB blocks in the copolymer chain. They are defined as:

$$< n_{\mathrm{A}} > = \sum iI[B(A)_i B] \Big/ \sum I[B(A)_i B] \tag{10.25}$$

$$< n_B > = \sum iI[A(B)_i A] \Big/ \sum iI[A(B)_i A], \tag{10.26}$$

where $I[B(A)_i B]$ and $I[A(B)_i A]$ are the molar fractions of oligomers $B(A)_i B$ and $A(B)_i A$, respectively. For the ultimate model, the number-average lengths are given by:

$$< n_{\mathrm{A}} >= 1/(1 - P_{\mathrm{AA}}), < n_{\mathrm{B}} >= 1/(1 - P_{\mathrm{BB}}) \tag{10.27}$$

For the penultimate model, the number-average lengths are given by:

$$< n_{\mathrm{A}} >= 1/(1 - Q_{\mathrm{AAA}}), < n_{\mathrm{B}} >= 1/(1 - Q_{\mathrm{BBB}}), \tag{10.28}$$

where Q_{AAA} and Q_{BBB} are **Q**-matrix elements.

Taking the averages in a different manner, one can define the weight-average block lengths $< SL_{\mathrm{A}} >$ and $< SL_{\mathrm{B}} >$. Wilczek-Vera et al. [29, 30] measured $< SL_{\mathrm{A}} >$ and $< SL_{\mathrm{B}} >$.

10.8 The Pen-Penultimate Model

The pen-penultimate model (also referred to as the ante-penultimate model) was proposed in the early 1970s [31, 32]. There are 16 different propagation reactions which can be written in a compact form:

$$\sim \underline{M}_p \underline{M}_q \underline{M}_r + M_s \overset{k_{pqrs}}{\rightarrow} \sim \underline{M}_p \underline{M}_q \underline{M}_r \underline{M}_s, \tag{10.29}$$

where $p, q, r, s = 1, 2$. In this case, it is very useful to define eight reactivity ratios $r_\mathrm{A}, r'_\mathrm{A}, r''_\mathrm{A}, r'''_\mathrm{A}, r_\mathrm{B}, r'_\mathrm{B}, r''_\mathrm{B}$ and r'''_B. The instantaneous composition equation for this model is:

$$c_\mathrm{A} = \Phi_\mathrm{antep}(f_\mathrm{A}, r_\mathrm{A}, r'_\mathrm{A}, r''_\mathrm{A}, r'''_\mathrm{A}, r_\mathrm{B}, r'_\mathrm{B}, r''_\mathrm{B}, r'''_\mathrm{B}) \tag{10.30}$$

An exact expression for Φ_antep has been derived [32]. The sequence is described by a third-order Markov chain with an associated **T**-matrix. The relations between the 16 rate constants k_{pqrs} in (10.29) and the Markov **T**-matrix elements are highly nonlinear relations. Nevertheless, using the eight reactivity ratios, they become almost linear [32]. The model suffers of a most evident limitation, namely too many degrees of freedom. Park et al. developed a simplified pen-penultimate model and used it for NMR sequencing of ethylene-norbornene copolymers [33].

10.9 The Complex Participation Model

In some processes, such as the free-radical copolymerization of maleic anhydride and styrene [34] or acrylonitrile and styrene [35,36], a peculiar effect arises, namely the formation of donor–acceptor complexes. These complexes are hold together by bonds which are very different form standard covalent bonds, and therefore they will be denoted as $M_1 \overset{\cdots}{\ldots} M_2$. In most cases, M_1 is electron-rich and M_2 is electron-poor. Four additional rate constants of propagation, $k^*_{11}, k^*_{12}, k^*_{21}, k^*_{22}$, are required to describe how the donor–acceptor complex is inglobated into the growing chain. As a result, the complex participation model kinetic scheme is made of eight equations. The molar fraction of A units in the copolymer is given by:

$$c_\mathrm{A} = \Phi_\mathrm{partic}(f_\mathrm{A}, k_{11}, k_{12}, k_{21}, k_{22}, k^*_{11}, k^*_{12}, k^*_{21}, k^*_{22}), \tag{10.31}$$

where Φ_partic denotes a quite involved functional dependence which is discussed in some detail in Kuchanov's review. Seiner and Litt used the complex participation model for NMR sequencing of methyl acrylate-diphenylethylene [37] and styrene-tetrachlorocyclopropene [38]. The consistent kinetic analysis [39] of the copolymerization (with the simultaneous occurrence of all the above reactions) leads to the conclusion that the probabilities of the sequences of the monomer units $\underline{M}_1$ and $\underline{M}_2$ in the macromolecules cannot be described by a Markov chain of any finite order. Consequently, in this very case, we deal with non-Markovian copolymers the general theory for which is not yet available. Since this conclusion has important mathematical consequences, some details of the derivation are in place. The authors [39] introduce a conditional coloring procedure and they distinguish monomer units as colored black or white. When this simple procedure is applied to sequences of monomer units, it allows to distinguish, besides their type, the manner of their adding to a polymer chain. The unit $\underline{M}_i$ is black if the corresponding monomer M_i is added to the radical as the first monomer of the complex. The unit $\underline{M}_i$ is white in all the other cases, namely when the corresponding monomer M_i is either added alone or as the second monomer of the complex. Thus, the states of the monomer unit are characterized by two features, namely its type ($i = 1, 2$) and its color (white or black):

$$S_1 \sim \underline{M}_1^{\text{whi}}, S_2 \sim \underline{M}_1^{\text{bla}}, S_3 \sim \underline{M}_2^{\text{whi}}, S_4 \sim \underline{M}_2^{\text{bla}} \tag{10.32}$$

In practice, the authors [39] find a general formula which predicts the occurrence of any kind of sequence in the copolymer. Then, they compare it with Markov chain and the comparison gives no match. The logical conclusion is that the copolymer sequence is non-Markovian.

10.10 ABC Copolymers and ABCD Copolymers

In 1944–1945, researchers realized that the terminal model can be modified to account for the presence of a z different monomers in the feed. There are z^2 different propagation reactions and $(z^2 - z)$ reactivity ratios. In ABCD copolymers, macromolecular chains of the type $A_p B_q C_r D_s$ are found. The average molar fractions of A, B, C, D units in the feed and in the copolymer are f_{A}, f_{B}, f_{C}, f_{D}, and c_{A}, c_{B}, c_{C}, c_{D}, respectively. In ABC copolymers, there are three different monomers in the feed. The terminal model predicts that the sequence of ABC copolymers is described by a first-order Markov chain with an associated 3 × 3 **P**-matrix. Matrix elements are given by an equation similar to (10.12):

$$P_{\text{AA}} = k_{11} f_{\text{A}} g_{3a}, \tag{10.33}$$

where $g_{3a} = k_{11} f_{\text{A}} + k_{12} f_{\text{B}} + k_{13} f_{\text{C}}$. Formulas for the other **P**-matrix elements are easily derived. Brar and Hekmatyar [40] used the above formula for NMR sequencing of an acrylonitrile-styrene-methylmethacrylate terpolymer. Ishigure et al. [41] used it for NMR sequencing of tetrafluoroethylene- propylene-isobutylene.

In ABCD copolymers, the sequence is described by a first-order Markov chain with an associated 4 × 4 **P**-matrix. Matrix elements are given by:

$$P_{\text{AA}} = k_{11} f_{\text{A}} g_{4a}, \tag{10.34}$$

where $g_{4a} = k_{11} f_{\text{A}} + k_{12} f_{\text{B}} + k_{13} f_{\text{C}} + k_{14} f_{\text{D}}$. Roland and Cheng [42] applied the above model to a styrene-methylmethacrylate-vinylidene chloride-acrylonitrile tetrapolymer, whereas Chen et al. [43] applied the model to NMR sequencing of an acrylic tetrapolymer referred to as Poly(IBMA-MMA-MAA-TBMA). Many authors derived mathematical procedures leading to formulas for the instantaneous bivariate distribution of chain lengths and compositions and for the instantaneous sequence distribution. Kuchanov's review reports all the cited mathematical procedures, along with a thorough discussion. The instantaneous sequence distribution can be cast in a closed form. For instance, the number-average block lengths $< n_\mathrm{A}$, $< n_\mathrm{B} >$, $< n_\mathrm{C} >$, $< n_\mathrm{D} >$ are defined by formulas similar to (10.25) and (10.26). The latter are given by the ratio of two geometrical series. Summing them and simplifying, the number-average block lengths, $< n_\mathrm{C} >$ and $< n_\mathrm{D} >$ become:

$$< n_\mathrm{C} >= \frac{1}{1 - P_\mathrm{CC}} < n_\mathrm{D} >= \frac{1}{1 - P_\mathrm{DD}} \tag{10.35}$$

The availability of simple algebraic expressions put in question the expediency of the application of the Monte Carlo method to compute the above values [44–50].

When $z > 4$, it becomes unpractical not to use the vector notation. The feed and the composition are simply $\mathbf{f_{ave}}$ and $\mathbf{c_{ave}}$. When $z = 7$, the components of $\mathbf{c_{ave}}$ are $(c_\mathrm{A}, c_\mathrm{B}, c_\mathrm{C}, c_\mathrm{D}, c_\mathrm{E}, c_\mathrm{F}, c_\mathrm{G})$. The bivariate distribution is a hyper-gaussian in z dimensions centered at the average composition vector.

10.11 Copolymerization at High Conversion

Free-radical copolymerization is often conducted in batch, up to high Ψ values (Ψ is the monomer-to-polymer conversion). The theory predicts that the resulting copolymer will be a mixture of the copolymer produced at time t_1, t_2, t_3, etc. More specifically, at time $t = t_\mathrm{nomore}$, the abundance of MPREF (the monomer which is preferentially included in the copolymer chain) becomes zero. The reaction medium becomes dense, the reaction accelerates (Trommdsdorff effect) and thus the sum $k_\mathrm{ptot} = k_{11} + k_{12} + k_{21} + k_{22}$ of the rate constants of propagation in (10.11) increases by an order of magnitude. The theory predicts that the compositional distribution histogram in (10.16) will loose its symmetry, it will become skewed and it will be at least a factor two or three wider than the Stockmayer's value in (10.21). With such values of the CODIHI's variance, it is absolutely apparent that copolymerization at high conversion cannot be described by a Markovian chain of any order. In fact, the feed f_A will change with time, and thus copolymer chains produced at different time will have different compositions. The system is governed by a set of ordinary differential equations. Kuchanov studied them in some detail and reported the results in his review. The composition can be represented as a point in a $(z-1)$ dimensional phase space. It is possible to follow its changes by drawing their trajectories in a multidimensional phase space. In the case of ABC copolymers, the space is a triangle with unit-length sides.

The operation of averaging a quantity over conversion will be denoted with a hat:

$$\widehat{c_\mathrm{A}} = \frac{1}{\Psi} \int_0^{\Psi} c_\mathrm{A} \mathrm{d}\Psi\prime \tag{10.36}$$

The integration of the instantaneous composition over reaction times can be performed symbolically and the result can be cast in a closed form. However, the resulting formula is not frequently used. Thus, numerical integration is preferred. The averaging operation can be performed on other quantities. Let us consider, for instance, the quantity $\widehat{I_{\mathrm{AAA}}}$ obtained integrating I_{AAA} (the molar fraction of AAA triads, which appears in (10.1) when XXXX = AAA) over conversions. $\widehat{I_{\mathrm{AAA}}}$ is given by an equation similar to (10.36). It is implicit and it would be extremely interesting to explicit it. Unfortunately, this has never been done. In other words, the sequence resulting from copolymerization at high conversion is unknown. In a paper by Stejskal et al. [51], the experimental conversion dependence of the average copolymer composition, $\widehat{c_{\mathrm{A}}}$, was approximated by a polynomial of nth degree using the least-squares method:

$$\widehat{c_{\mathrm{A}}} = \sum_{k=0}^{n} c_k \Psi^k, \qquad (10.37)$$

where c_k are the kth polynomial coefficients. In the light of the above difficulties, the fact that many authors developed computer programs which evaluate the sequence distribution at high conversion [52] is not surprising. For instance, Ray and coworkers (Wisconsin University) developed a computer program called POLYRED which gives state-of-the-art results.

If the feed is extremely unbalanced towards one of the two monomers and this monomer is MPREF, a very simple model [53] for the sequence distribution holds, based on the assumption that $f_{\mathrm{A}}(t)$ can be approximated by a step function. Let us suppose that initially $f_{\mathrm{A}} = 0.90$. Then, $f_{\mathrm{A}}(t) = 0.90$ if $t < t_{\mathrm{nomore}}$ and $f_{\mathrm{A}}(t) = 1$ if $t \geq t_{\mathrm{nomore}}$. In this case, at $t = t_{\mathrm{nomore}}$, the reaction produces pure homopolymer chains and the sequence is the sum of low-conversion sequence plus homopolymer. The CODIHI becomes:

$$\mathrm{CODIHI} = (1 - a_{\mathrm{cop}})\mathrm{HOMO} + a_{\mathrm{cop}} \int W_s(y)\mathrm{d}s', \qquad (10.38)$$

where a_{cop} is the abundance of copolymer produced at time $t = t_{\mathrm{nomore}}$.

For chemical industries which produce and sell copolymers, the presence of homopolymer is unwanted. In order to avoid it, when Ψ reaches a given value, Ψ_{vessel} (usually a value of 0.20–0.30), the reaction vessel is opened and a large amount, α_{MPREF}, of MPREF is poured inside the vessel.

10.12 Preferential Sorption

During the experimental investigation of the free-radical copolymerization of styrene and methacrylic acid at low-conversion, a number of peculiarities were noted. The growing radical (actually a macroradical) was not a coil (as usual) and it was capable of creating a microenvironment. Due to its globular state, the monomer mixture composition inside and outside the globule were markedly different. Such partitioning of monomers is often referred to as the Harwood Bootstrap effect. A model has been proposed which accounts for preferential sorption [54]. The predicted sequence is obtained by averaging over conversions (10.36). This is needed even when conversion is low ($\Psi < 0.10$). The model predicts a broad CODIHI.

10.13 Equilibrium Copolymerization

Equilibrium copolymerization was experimentally observed in various systems: for instance, in a system were the two monomers were tetrahydrofuran and 3-methyltetrahydrofuran [55], in a system were the two monomers were propylene sulphide and elemental sulphur (actually the octamer, S_8) [56] and in various systems containing alpha methylstyrene [57]. In equilibrium copolymerization, the rate of depolymerization (the reaction in which the chain shortens) is not negligibly small. This reaction is extremely attractive because it can be modeled and the model can be solved exactly (in a closed form). This intrinsic elegance was clear from the very beginning (late 1960s). The initiator can react in three different ways and thus we have three cases, namely case I, case II and case III. Yan and coworkers used induction to solve the set of equations which gives the weight of chain $A_m B_n$ [58]. Thereafter, they considered the case in which the rate of monomer addition depends on the last unit of the macromolecular chain. This turned out to be much more complex. They first derived a general result [59] in an implicit form and then they derived the explicit solutions for case I, case II and case III [60].

10.14 Emulsion Copolymerization

In emulsion copolymerization, a surfactant such as sodium dodecyl sulphate is used to produce micelles. The Smith–Ewart theory explains the distinct features of the MMD. The copolymer sequence is identical to the sequence obtained with free-radical copolymerization [61].

10.15 Copolymerization by the Polymer Analog Reaction

When the reaction which transforms a prepolymer into a polymer is stopped at medium yield, a copolymer is formed. For instance, when poly(styrene) is reacted with sulphuric acid for short times, a copolymer with units of styrene sulfonic acid and styrene is obtained. In a similar manner, when the hydrolysis of poly(vinyl acetate) is stopped at medium conversion, a random copolymer with units of vinyl alcohol and vinyl acetate is formed. These processes are industrially important (e.g. for ion-exchange resins, for water-soluble polymers, for electrically conductive polymers, for carbon fibers form polyacrylonitrile, for polymethylmethacrylate, for cosmetics). They are usually referred to as polymer analog reactions or polymer analogous reactions [97]. The sequence of these copolymers can be modeled, as discussed in a lengthy chapter by Kuchanov in the book on contemporary chemistry. The case of Poly(vinyl chloride) reacted with zinc is slightly different, since it involves two consecutive vinyl chloride units [62]. For this reason, the sequence cannot be described by a Markov chain in $\underline{M}_1$ and $\underline{M}_2$. An exact solution to this non-markovian problem has been derived by the Nobel-prize winner Paul Flory. He derived a relation between the abundance $A(n+1)$, $A(n)$, $A(n-1)$ of chains which suffer $(n+1)$, (n), $(n-1)$ attacks, respectively. The prediction is that the reaction does not go to completion. Zinc cannot "chew away" all chlorine from the polymer since some vinyl-chloride units become immune to Zinc.

10.16 Anionic Copolymerization

Anionic copolymerization proceeds via metal-organic sites such as butyl-lithium. When the two monomers are dissolved in an aprotic solvent, and neither transfer nor any kind of termination is possible, the copolymerization will proceed until all the monomer is consumed. The copolymerization is *living* and a further addition of monomer will result in further growth of the chains. Anionic copolymerization at constant monomer ratio was considered. Equation (10.7) still holds and this implies that the difference $|c_\mathrm{A} - f_\mathrm{A}|$ initially grows. However, at high Ψ values it becomes small. The set of differential equations was transformed in a linear set of equations (using the Laplace transform and Gauss hypergeometric function) and then solved [63,64]. Anionic copolymerization lends itself to the production of tri-penta-tapered copolymers and, more in general, of gradient copolymers (already mentioned in the introduction). Two reviews on gradient copolymers appeared [65,66]. The review by Beginn [66] makes a distiction between spontaneous gradient and forced gradient. The sequence is fascinating [67].

10.17 Block Copolymerization

Copolymers with long AAAA and BBBB blocks are easily obtained in anionic copolymerization reactions by changing (in a drastic manner) the monomer ratio. The initial feed is made entirely of monomer A. It starts polymerizing and it goes on until the monomer is consumed. At this point, monomer B is added. This procedure is referred to as *sequential* anionic copolymerization. If one compares the MMD for the copolymer and the generating homopolymer, one finds that both MMDs are very narrow ($\bar{\prod}_w/\bar{\prod}_n < 1.15$). Furthermore, a *narrowing effect* [68] takes place and the latter is broader, despite its short length. A model for this reaction has been put forward [69] which predicts all copolymers properties, including the shape of the bivariate distribution. The model makes use of the product of two Schulz–Zimm distributions:

$$W_s(y) = c_{11}\mathrm{MMD}_1(c_{\mathrm{A}}s)\mathrm{MMD}_2(c_{\mathrm{B}}s), \tag{10.39}$$

where c_{11} is a normalization factor given by the product of a decreasing exponential and the Kummer confluent hypergeometric function, where $\mathrm{MMD}_k(s) = s^{\alpha_k}\exp(-\theta_k s)$ with $k = 1, 2$, and θ and α are parameters connected with the length of the two blocks. It is quite apparent that the sequence distribution in (10.39) can be roughly approximated by a Markov chain of high order (say tenth order) in which one of the transition probabilities is 0.99999 and its complementary is 0.00001. In fact, the stationary vector of such Markov chain possesses long AAAA and BBBB blocks. However, it is also apparent that the predicted sequence cannot be described exactly by a Markov chain of any order. In fact, in sequential anionic copolymerization, triblock, tetrablock and multiblock sequences are not produced (their probability is exactly zero). On the other hand, Markov chain are probabilistic and thus the probability of triblock, tetrablock and multiblock sequences is never exactly zero.

The model in (10.39) has a very strong limitation. It cannot describe a very common case, i.e., when, after the copolymerization, some unreacted homopolymer chains are left. In order to circumvent this obstacle, an alternative model has recently been proposed in which K, an integral operator kernel, acts on a set of transient states constituting continuum [70]. Each copolymer chain is made of a number of elementary chains formed at different instants $\tau_1, \tau_2, \cdots$. These elementary chains vary in composition due to the fact that the feed is different at time $\tau_1, \tau_2, \cdots$. The author resorts to a "labeling trick" which consists in marking each elementary chain by the time of its formation.

10.18 Copolymerization Using Metallocene Catalysts

A metallocene is a compound with the general formula $(C_5H_5) - \mathrm{Me} - (C_5H_5)$ consisting of two cyclopentadienyl anions (Cp) bound to a metal center (Me) in the oxidation state II. Metallocene derivatives with Zirconium (or another atom belonging to group 4) catalyze olefin polymerization. The catalyst is *multisite*. As a consequence, the resulting copolymer is a mixture of copolymers formed at site 1, 2, 3,,, etc. The sequence distribution $I^{\mathrm{tot}}_{\mathrm{XXXX}}$ is obtained summing all the components of the mixture:

$$I^{\mathrm{tot}}_{\mathrm{XXXX}} = I^{(1)}_{\mathrm{XXXX}} + I^{(2)}_{\mathrm{XXXX}} + I^{(3)}_{\mathrm{XXXX}} + \ldots \tag{10.40}$$

Seger et al. [71] applied the above model to poly(ethylene-co-1-hexene). The case of two markoffian chains of zero-th order has been studied in detail. Two computer programs called EXCO.TETRAD and EXO.PENTAD have been used in these studies [72].

10.19 Copolymerization Followed by Partial Degradation

Chemical reactions capable of cleaving the bonds that hold together the repeat units are called degradation reactions. At completion, the degradation process yields methane, propane, methanol, butanol, CO^2 or other small molecules. When the reaction products are a mixture of dimers, trimers, tetramers and other short $A_m B_n$ oligomers, the degradation is not complete, and we refer this reaction as *partial degradation*. The degradation process is referred as *non selective* when A units and B units are attacked indiscriminately. The degradation is referred to as *totally selective* when only one of the two units (A or B) is attacked and the other unit remains untouched by the degradation process. The sequence of the resulting copolymer depends on the reaction used to synthetize the copolymer and also on the partial degradation reaction [73]. The molar fraction, $I_{A_m B_n}$, of the oligomer $A_m B_n$ is given by:

$$I_{A_m B_n} = \Phi_1(s, c_{\mathrm{A}}), \tag{10.41}$$

where Φ_1 denotes generic dependence. Formulas for very short oligomers (dimers and trimers) were derived long ago, and they turned out to be very useful for the experimental work on gas chromatography (in fact dimers and trimers can be easily brought in the gas phase). Formulas for longer oligomers were derived by our group in the early 1990s. The reason is connected with the fact that we used mass spectrometers for sequence determination, and many MS instruments (at that time) could not produce and detect ions above a certain mass limit. Using partial degradation, we were able to reduce the chain length below the mass limit. For instance, using (10.41) with $s = 5, 6, 7, 8, 9, 10$, we sequenced by MS an acrylonitrile-styrene copolymer subjected to partial degradation by pyrolysis and a styrene-butadiene subjected to partial degradation by ozonolysis. An entire collection of formulas has been published [73].

10.20 Copolymerization with Sequence Constrains

In some copolymerization and terpolymerization reactions, the sequence AA or BB or ABB is not formed. For instance, maleic anhydride (an industrially important monomer) does not homopropagate (at least under usual free-radical conditions). However, if a second monomer is added, the copolymerization reaction produces a random copolymer. It is actually a pseudo-random copolymer, which differs from a random one, due to the systematic absence of two consecutive maleic anhydride units. The equations which define the sequence distribution are always the same: the terminal model equation and the penultimate model equations. Simply, some parameters are zero. Some authors obtained a pseudo-random copolymer reacting sulphur dioxide with styrene, and they succeeded in performing NMR sequencing of the copolymer using an ultra-simplified pen-penultimate model [74]. The simplification was based on the fact that sulphur dioxide cannot homopolymerize. Braun et al. analysed by NMR the products of the terpolymerization of three nonhomopolymerizable monomers, namely N-ethylmaleimide, anethol, and trans-stilbene. This reaction is rather involved, since donor–acceptor complexes may form. They carefully assigned peaks in the NMR spectrum and estimated the sequence distribution [75]. They also reported another case with chloroethyl vinyl ether [76].

In some copolymerization reactions, an exact-sequence copolymer is produced, instead of a pseudo-random copolymer. Although exactly alternating AB copolymers are by far the most common, other types of exact-sequence copolymers have produced, such as the copolymer $(\mathrm{AAB})_n$. Upon re-definition of the repeat unit, exact-sequence copolymers are transformed into homopolymers. In our lab, mass spectrometry is used for copolymer analysis, and a computational procedure was developed specifically for exact-sequence copolymers [77]. A very long artificial chain is generated (the length is Z_{long}) and a procedure called `find substring in string` is used. The molar fraction, $I_{A_m B_n}$, of the oligomer $A_m B_n$ is given by:

$$I_{A_m B_n} = \Phi_2 / Z_{\text{long}}, \tag{10.42}$$

where Φ_2 is the number of times the sequence XXXX, characterized by a number of A and B units compatible with $A_m B_n$, appears in the artificial chain. It is not clear whether an algebraic solution of (10.42) can be found.

10.21 The Perturbed Markovian Model

The first-order and second-order markovian are not flexible sequence distributions. For instance, the CODIHI is invariably narrow. The *perturbed markovian* [78] model, instead, is based on first-order markovian but it is much more flexible. One assumes that the **P**-matrix elements take on a range of values, instead of specific values. Let $\pm\varepsilon$ be the range of values that the reaction probabilities may take on. Then:

$$P_{\text{AA}} = \langle P_{\text{AA}} \rangle \pm \varepsilon,\ P_{\text{AB}} = \langle P_{\text{AB}} \rangle \mp \varepsilon \tag{10.43}$$

$$P_{\text{BA}} = \langle P_{\text{BA}} \rangle \pm \varepsilon,\ P_{\text{BB}} = \langle P_{\text{BB}} \rangle \mp \varepsilon \tag{10.44}$$

The abundances for diads, triads, tetrads, etc. are given by:

$$I_{\text{XXXX}} = I_{\text{XXXX}}^{\text{unpert}} + H_{\text{pert}}, \tag{10.45}$$

where $I_{\text{XXXX}}^{\text{unpert}} = \langle P_{\text{AA}} \rangle^{n1} \langle P_{\text{AB}} \rangle^{n2} \langle P_{\text{BA}} \rangle^{n3} \langle P_{\text{BB}} \rangle^{n4}$ and where the additional term H_{pert} contains ε. This implies that perturbed markovian is not markovian. Contrary to its name, it is based on different build-up rules and thus it is a non-markovian sequence distribution. Another *perturbed markovian* [79] model was proposed, which has a larger number of degrees of freedom (four). It uses an asymmetric function, namely an exponentially modified gaussian, and therefore, the average **P**-matrix elements and unperturbed **P**-matrix elements are different, contrary to (10.44).

A wide variety copolymerization reactions can be modeled; for instance, the copolymerization of styrene and ethyl methacrylate using three different polymerization techniques, the copolymerization of styrene and butyl acrylate in batch mode (i.e., when the reaction vessel is not opened during the reaction), free-radically polymerized vinyl chloride-vinylidene chloride copolymers and isobutylene/isoprene copolymers obtained by cationic copolymerization at different Ψ values (Ψ is the monomer-to-polymer conversion). It can be applied also to homopolymers. For instance in the anionic copolymerization of poly(methyl-alfa chloroacrylate), tacticity is treated as a copolymer problem, with meso (m) and racemic (r) configurations as comonomers.

10.22 Sequence Measurement

The measurement of the sequence distribution of addition polymers can be performed using pryolysis-gas chromatography (Py-GC), liquid chromatography, wet chemistry methods, radiolabeling and spectroscopy [80]. Three popular spectroscopic techniques are FTIR, NMR and MS. The first technique (FTIR) is very important for assessing copolymer composition. It often happens that two bands are detected in the FTIR spectrum, due to A and B units, respectively. The composition is obtained by measuring the areas under the two bands and taking the ratio [81]. Nevertheless, FTIR is less suited than NMR and MS for copolymer sequencing, since the signals in the FTIR spectrum are due to short sequences, shorter than the sequences "seen" by NMR and MS. For this reason, it will not be discussed.

10.23 Nuclear Magnetic Resonance

Coleman and Fox published a famous paper on *Journal of Polymer Science*, dealing with stationary sequences and NMR [82]. The paper is peculiar. A series of theorems are proven in a manner similar to math handbooks, starting with some definitions, two statements (the hypothesis and the thesis) and the formal proof. After these pioneering studies, many experiments were conducted which showed that copolymer sequencing by NMR is feasible. Frank Bovey analysed a number of copolymers by NMR. He also noted that the natural abundance of fluorine isotopes is very attractive and he worked on repeat units which possess one, two, three, four fluorine atoms. In the late 1970s, the field was fully mature and Randall summarized the know-how and the entire methodology in a book [83]. It has an initial chapter that deals with NMR spectra and spectral assignments, followed by a chapter on number-average sequence lengths in vinyl homopolymers (tacticity, meso and racemic additions) and a chapter on number-average sequence lengths in copolymers and terpolymers. The formulas that give the number-average sequence lengths are linear combinations of NMR intensities $I_1, I_2, I_3 ...$ of the type

$$< n_{\mathrm{A}} >= \frac{a_{11}I_1 + a_{12}I_2 + a_{13}I_3 + ...}{a_{21}I_1 + a_{22}I_2 + a_{23}I_3 + ...}, \tag{10.46}$$

where $a_{11}, a_{12}, a_{13} ...$ are integer numbers. It may happen that two NMR signals are not resolved. This is disappointing, since (10.46) requires a value and *skipping* the value is forbidden. This is the reason why a chapter of the book deals with statistical analysis. In fact, assuming a statistical model and performing a best-fit minimization on SS (the sum of squares), the deconvolution of overlapping peaks can be performed. When fitting NMR spectral data with models that have different numbers of degrees of freedom, SS cannot be used as a measure of the goodness-of-fit. The reader should bear in mind that when the complex-participation model and the penultimate model were first proposed, NMR was already the leading technique, and thus the words `comparison with experiment` and `comparison with the NMR spectrum` were synonymous. The book by Randall was published more that 30 years ago and, thus it cannot account for progress in the field. Unfortunately, a comprehensive updated review would require many volumes. On the other hand, books on NMR of polymers devote a section to copolymer sequencing. Here, we will just cite some papers to show that many new copolymer systems have been investigated, and that our understanding of known systems has become deeper. Llauro et al. [84] recorded high-quality NMR spectra of ethylene-butadiene copolymers and performed sequence analysis. German and coworkers [85] compared NMR spectra of alternating and random styrene-methyl methacrylate copolymers. Boggioni et al. [86] were able to assign each single peak which appears in the NMR spectrum of propene-norbornene copolymers. Segre et al. [87] used NMR to measure the sequence of optically active copolymers with units of methyl-l-hexene and styrene. Bailey et al. [88] used different sequence distribution models to fit the NMR spectra of copolymers with units of ethylene and octene and compared the results. Busico et al. [89] analysed propylene by NMR and pentads, heptads, nonads, up to 11-ads were seen as separate signals. VanDoremaele et al. [90] recorded the NMR spectra of styrene-methyl acrylate copolymers obtained in emulsion using *n*-dodecyl mercaptan as detergent. Randall and Ruff [91] compared the NMR spectra of ethylene-1-butene, ethylene-1-hexene, and ethylene-1-octene copolymers and noted some common features and general trends. Kraemer et al. [92] analysed a series of copolymer with units of styrene and units of ethyl acrylate by on-line coupled SEC-NMR. They discovered that the compositional distribution histogram (CODIHI, see (10.16)) is large, especially when the feed is rich in ethyl acrylate.

Apart from classical proton and carbon NMR, one can perform two-dimensional experiments and also distortionless enhancement by polarization transfer (DEPT). Figure 10.4 reports the DEPT-NMR spectrum of two styrene-maleic anhydride copolymers referred to as S78 and S91 [93]. The styrene-centered triads MSM, SSM+MSS and SSS are seen at 35, 40 and 45 ppm respectively. The area under

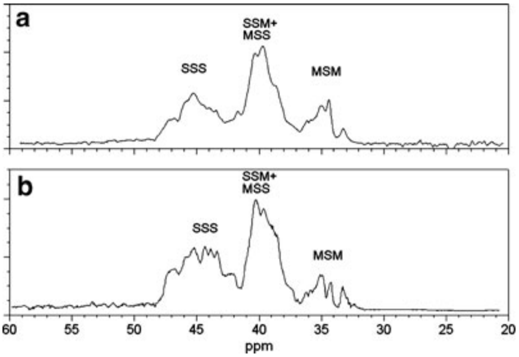


Fig. 10.4 DEPT-NMR spectrum of a two styrene-maleic copolymers. Referred to as 578 (**a**) and 591 (**b**). Reprinted from [93] with permission from the copyright holder ACS 2001

the three peaks reflects the abundance of triads, which in turn, reflects the sequence distribution. Using the relationship between molar fractions and weight fractions, the variance of the compositional distribution histogram in (10.17) turns out to be $\sigma^2 = 0.0844$ for sample S78 and $\sigma^2 = 0.0676$ for sample S91. The theoretical variances for S78 and S91 are $\sigma^2 = 0.015$ and $\sigma^2 = 0.0097$ respectively, which are almost an order of magnitude lower than the former ones.

10.24 Mass Spectrometry

In 2002, a review appeared which deals with MS and copolymers [94], and the use of MS for copolymer sequencing is described using a number of examples. Mass spectra of copolymers are rich in peaks. As a matter of fact, the number of peaks grows as the mass grows. This wealth of information is pleasant for copolymer sequencing. Mass spectra are usually processed in two steps [94]. In a first (preliminary) step, trivial sequential data are extracted from mass spectra using paper-and-pencil calculation (we refer to this step as the *composition estimates*). In a second step, a computer is used to derive the full copolymer sequence. A computer program called MACO was developed by us.

Some mass spectrometers are equipped with a fast atom bombardment (FAB) source. Figure 10.5 reports the structure of a copolymer in which styrene (ST) and butadiene (BU) units are found along the chain. The composition is 85/15. The copolymer was subjected to partial degradation by ozonolysis for 10, 30, 60, 120 min and the reaction products were labelled OZ10, OZ30, OZ60, OZ120. The figure also reports the FAB mass spectrum of OZ60 [95]. Peaks at $m/z = 500$, 554, 608, 662, 716, 770, 824 are due to oligomers of the type St_3BU_n, where $n = 0, 1, 2, 3, 4, 5, 6$. The length of styrene-butadiene chains is reduced by ozonolysis. In the nomenclature introduced with (10.41), ozonolysis is a *totally selective* cleavage since only butadiene units are attacked by ozone. The sequence of the undegraded polymer must be reconstructed from the knowledge of the sequence of the partially degraded ones (OZ10, OZ30, OZ60, OZ120). The computer program

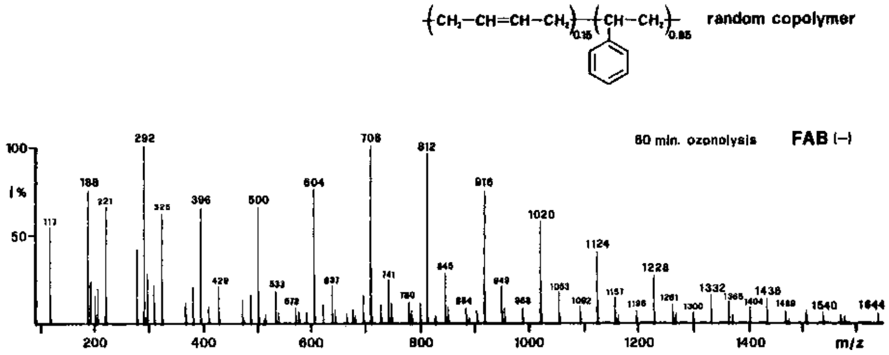


Fig. 10.5 FAB mass spectrum of a copolymer in which styrene (ST) and butadiene (BU) units are found along the chain, The copolymer was subjected to ozonolysis for 60 min. Reprinted from [95], with permission from the copyright holder ACS 1991

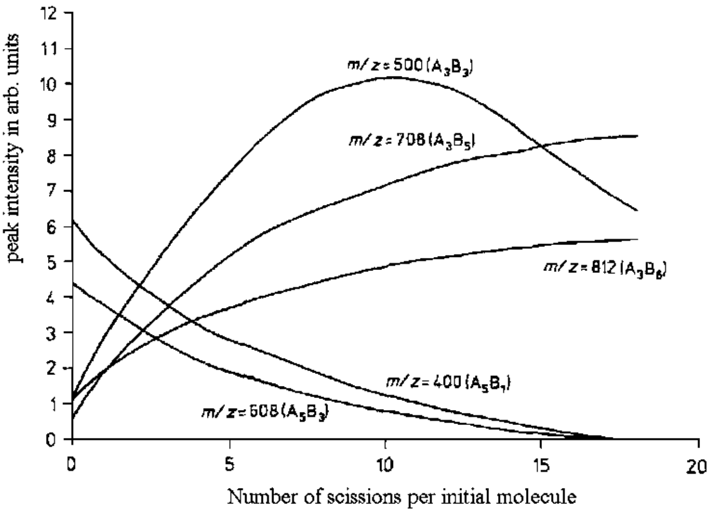


Fig. 10.6 Results of a copolymer scission reaction simulation: the intensity variations of five mass spectral peaks as the average number of scissions per initial molecule (S) varies from 0 to 18. Reprinted from [96] with permission from the copyright holder Wiley 1993

called MACO was equipped with a new subroutine (ad hoc) which can simulate the entire ozonolysis process, including the mass spectra of the partially degraded samples [96]. Figure 10.6 reports the intensity variations of five mass spectral peaks, as the average number of scissions per initial molecule (S) varies from 0 to 18. From the inspection of the figure, it can be seen that peaks at $m/z = 708$ and 812 are enhanced and peaks at $m/z = 608$ and 400 become weaker. The peak at $m/z = 500$ shows an initial increase, reaches a maximum at $S = 10$ and then decreases. The theory predicts that the MS intensities associated with short oligomers are enhanced with respect to MS intensities associated with long oligomers. As a matter of fact, some of the results in the figure are in agreement with this prediction. For instance, the intensity of the peak at $m/z = 708$ (corresponding to the octamer St_3BU_5) increases with respect to the intensity of the peak at $m/z = 812$ (corresponding to the nonamer St_3BU_6). The theoretical predictions were compared with experiment results and the agreement was good [96].

MALDI is another ion source used in MS and Fig. 10.7 reports the MALDI mass spectrum of a copolymer [97] with units of hydroxy butyrate (HB) and valerolactone (VL). It displays more than 600 MS peaks ranging from 1,500 to 4,000 Da. The peaks are due to the chains containing crotonate starting group, and desorb as potassium adducts as reported in Fig. 10.8. The peaks at $m/z = 1{,}341$, 1,441, 1,541, 1,642 and 1,742 are very rich in VL units. The mass spectral intensities of the 600 MS peaks were given as input to the MACO computer program and the result of the calculation was plotted. Figure 10.9 (solid line) reports the compositional distribution histogram, as derived from mass spectrum (the histogram bars are so close that it is actually a curve). The abscissa is c_{VL} the molar fraction of

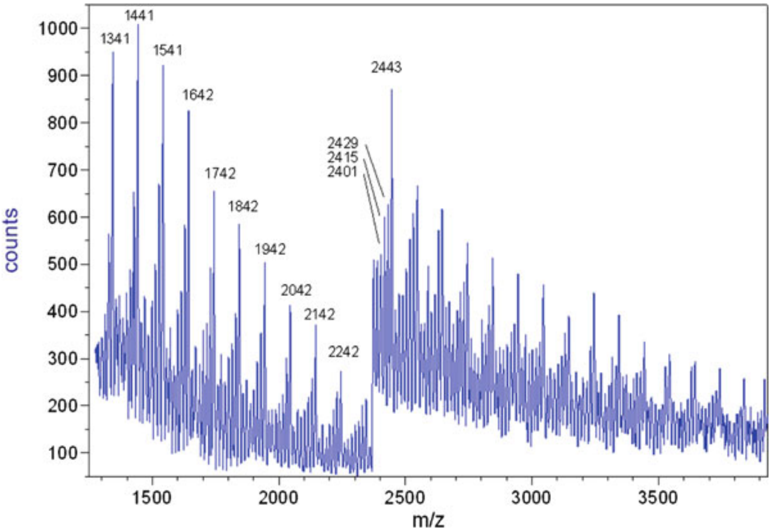


Fig. 10.7 MALDI mass spectrum of a copolymer with units of hydroxy butyrate (HB) and valerolactone (VL). Reprinted from [97] with permission from the copyright holder Wiley 2002

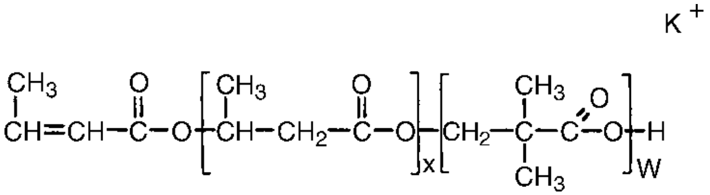


Fig. 10.8 Structure of the ions seen in the MALDI spectrum of the PVL-PHH copolymer

VL in the chain. The weight is initially high, it decreases, it touches a minimum at $c_{\mathrm{VL}} \approx 0.05$, then it grows, it reaches a maximum at $c_{\mathrm{VL}} \approx 0.22$ and then it decreases gently to zero. The same figure (dashed line) reports the compositional distribution curve predicted by (10.17), which will be referred to as the "old" model. The predicted curve can account for some of the experimental observations in the region where $c_{\mathrm{VL}} \in [0.30\text{–}0.10]$, but it describes incorrectly the experimentally observed behavior in the region where c_{VL} is below 0.10. Due to the bad results obtained with the old model, it is necessary to resort to a new model, in which the histogram is the sum of two "parts":

$$\mathrm{CODIHI} = \int W_s(y, h_1, h_2)\mathrm{d}s', \tag{10.47}$$

where h_1 and h_2 are two parameters. Figure 10.9 (dotted line) reports the compositional distribution curve predicted by (10.47), and it can be seen that the predicted curve agrees well with the experimental findings. It must be concluded that in this case, the new model performs much better than the old one.

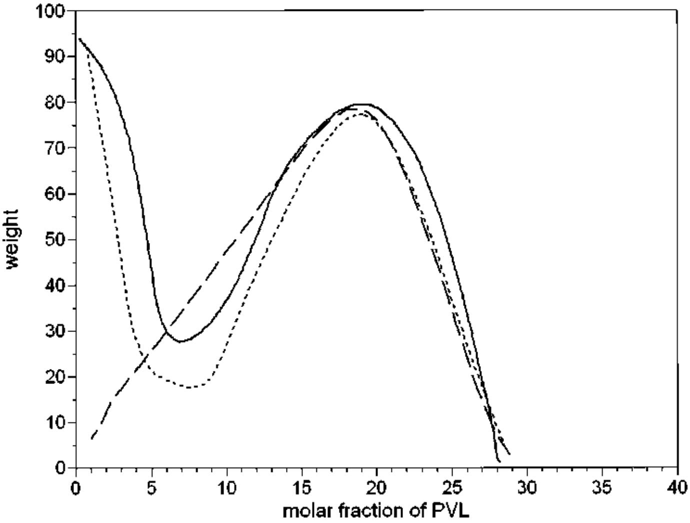


Fig. 10.9 CODIHI (compositional distribution histogram) for the copolymer with units of hydroxy butyrate (HB) and valerolactone (VL). The experimental CODIHI (*solid line*) is compared with the prediction of the old model (*dashed line*) and the new model (*dotted line*). Reprinted from [97]. with permission from the copyright holder Wiley 2002

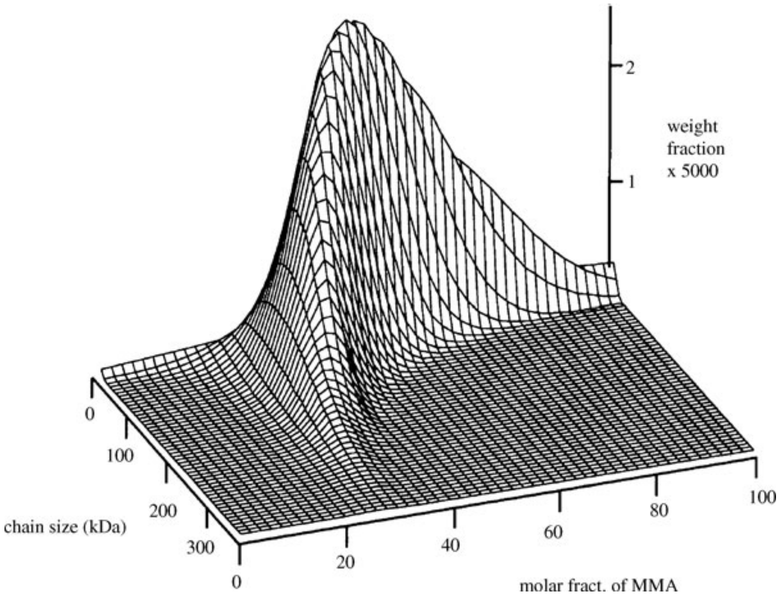


Fig. 10.10 Bivariate distribution of chain sizes and compositions for the MMA-BA copolymer Reprinted from [98] with permission from the copyright holder ACS 1999

The SEC–NMR–MALDI method consists in injecting a copolymer sample in a size-exclusion chromatography (SEC) apparatus, in collecting 60–90 fractions and in analyzing selected fractions by NMR and by MALDI. The SEC–NMR–MALDI method was applied to a copolymer with units of methyl methacrylate (MMA) and butyl-acrylate (BA for brief) reacted at high conversion. Figure 10.10 reports the bivariate distribution of chain sizes and compositions for the MMA-BA copolymer [98]. It can be seen that the average molar fraction of MMA in the copolymer, c_{MMA}, changes. At 300 kDa, $c_{\mathrm{MMA}} = 0.30$, whereas at 200 kDa, $c_{\mathrm{MMA}} = 0.35$. At lower masses c_{MMA} increases to 0.40, 0.50 and even higher values. The results reported in Fig. 10.10 demonstrate beyond any possible doubt that the SEC–NMR–MALDI method is very powerful.

References

1. P.C. Painter, M.M. Coleman, **Fundamentals of Polymer Science**, Technomic Publ., Lancaster, 1997
2. P. Rempp, E.W. Merril, **Polymer Synthesis, 2nd edition**, Huthig-Wepf, Basel, 1991
3. R.H. Boyd, P.J. Phillips, **The Science of Polymer Molecules**, Cambridge University Press, NY, 1993
4. G.G. Odian, **Principles of Polymerization, 4th edition**, page 437, Wiley-VCH, NY, 2004
5. S. Jouenne, J.A. Gonzalez-Leon, A.V. Ruzette, P. Lodefier, S. Tence-Girault, L. Leibler, Macromolecules 40, 2432–2442 (2007)
6. A. Zargar, F.J. Schork, Ind. Eng. Chem. Res. 48, 4245–4253 (2009)
7. M.R. Rivera, A.A. Rodriguez-Hernandez, N. Hernandez, P. Castillo, E. Saldivar, L. Rios, Ind. Eng. Chem. Res. 44, 2792–2801 (2005)
8. A. Krallis, D. Meimaroglou, C. Kiparissides, Chem. Eng. Sci. 63, 4342–4360 (2008)
9. A. Keramopoulos, C. Kiparissides, Macromolecules 35, 4155–4166 (2002)
10. D.S. Achilias, C. Kiparissides, Polymer 35, 1714–1721 (1994)
11. S.I. Kuchanov, Adv. Polym. Sci. 103, 3–101 (1992)
12. A.S. Brar, K. Dutta, Macromol. Chem. Phys. 199, 2005–2015 (1998)
13. A.S. Brar, S. Charan, J. Polym. Sci. Part A Polym. Chem. 34, 333–339 (1996)
14. A.S. Brar, J. Kaur, Eur. Polym. J. 41 2278–2289 (2005)
15. A.S. Brar, A. Sunita, Eur. Polym. J. 27, 17–20 (1991)
16. S. Hooda, A.S. Brar, A.K. Goyal, J. Mol. Str. 828, 25–37 (2007)
17. J.C.J.F. Tacx, J.L. Ammerdorffer, A.L. German, Polymer 29, 2087–2095 (1988)
18. W.H. Stockmayer, J. Chem. Phys. 13, 199–207 (1945)
19. J.C.J.F. Tacx, H.N. Linssen, A.L. German, J. Polym. Sci. Part A Polym. Chem. 26, 61–69 (1988)
20. H.N. Cheng, S.B. Tam, L.J. Kasehagen, Macromolecules 25, 3779–3785 (1992)
21. R. Landry, A. Penlidis, T.A. Duever, J. Polym. Sci. A Polym. Chem. 38, 2319–2332 (2000)
22. S. Losio, P. Stagnaro, T. Motta, M.C. Sacchi, F. Piemontesi, M. Galimberti, Macromolecules 41, 1104–1111 (2008)
23. I. Tritto, L. Boggioni, J.C. Jansen, K. Thorshaug, M.C. Sacchi, D.R. Ferro, Macromolecules 35, 616–623 (2002)
24. K. Yamada, T. Nakano, Y. Okamoto, Macromolecules 31, 7598–7605 (1998)
25. N. Stribeck, Polymer 33, 2792–2795 (1992)
26. W.K. Czerwinski, Polymer 38, 1381–1385 (1997)
27. H.J. Harwood, W.M. Ritchey, J. Polym. Sci. Part B Polym. Lett. 2, 601–607 (1964)
28. W. Ring, J. Polym. Sci. Part B Polym. Lett. 1, 323–327 (1963)
29. G. Wilczek-Vera, P.O. Danis, A. Eisenberg, Macromolecules 29, 4036–4044 (1996)

30. G. Wilczek-Vera, Y. Yu, K. Waddell, P.O. Danis, A. Eisenberg, Rapid Commun. Mass Spectrom. 13, 764–777 (1999)
31. K. Ito, Y. Yamashita, J. Polym. Sci. Part A: Gen. Pap. 3, 2165–2187 (1965)
32. G.E. Ham, J. Macromol. Sci. Part A Pure Appl. Chem. 5, 453–458 (1971)
33. S.Y. Park, J. Lee, K.Y. Choi, Macromol. React. Eng. 1, 68–77 (2007)
34. R.E. Cais, R.G. Farmer, D.J.T. Hill, J.H. O'Donnell, P.W. O'Sulllvan, Ind. Eng. Chem. Prod. Res. Dev. 19, 412–415 (1980)
35. D.J.T. Hill, J.H. O'Donnell, P.W. O'Sullivan, Macromolecules 15, 960–966 (1982)
36. L.V. Medyakova, Z.M.O. Rzaev, A. Guner, G. Kibarer, J. Polym. Sci. Part A: Polym. Chem. 38 2652–2662 (2000)
37. M. Litt. J.A. Seiner, Macromolecules 4, 314–316 (1971)
38. M. Litt. J.A. Seiner, Macromolecules 4, 316–319 (1971)
39. S.I. Kuchanov, S.V. Korolev, V.P. Zubov, V.A. Kabanov, Polymer 25, 100–106 (1984)
40. A.S. Brar, S.K. Hekmatyar, J. Appl. Polym. Sci. 74, 3026–3032 (1999)
41. K. Ishigure, S. Watanabe, Y. Tabata, K. Oshima, Macromolecules 13, 1630–1634 (1980)
42. M.T. Roland, H.N. Cheng, Macromolecules 24 2015–2018 (1991)
43. W.C. Chen, Y. Chuang, W.Y. Chiu, J. Appl. Polym. Sci. 79, 853–863 (2001)
44. I. Motoc, S. Holban, R. Vancea, J. Polym. Sci. Polym. Chem. Edit. 16, 1601–1608 (1978)
45. I. Motoc, R. Vancea, S. Holban, J. Polym. Sci. Polym. Chem. Edit. 16, 1595–1599 (1978)
46. I. Motoc, S. Holban, D. Ciubotariu, J. Polym. Sci. Polym. Chem. Edit. 15, 1465–1472 (1977)
47. I. Motoc, I. Muscutariu, J. Macromol Sci, Part A Pure Appl. Chem. 15, 75–84 (1981)
48. I. Motoc, I. Muscutariu, S. Holban, O. Dragomir, J. Polym. Sci. Polym. Chem. Edit. 18, 1565–1575 (1980)
49. F.M. Mirabella, Polymer 18, 705–711 (1977)
50. F.M. Mirabella, Polymer 18, 925–929 (1977)
51. J. Stejskal, P. Kratochvil, D. Strakova, 0. Prochazka, Macromolecules 19, 1575–1589 (1986)
52. H.J. Harwood, J. Polym. Sci. Part C Polym. Symp. 25, 37–45 (1968)
53. J.C.J.R Tacx, J.L. Ammerdorffer, A.L. German, Polym. Bulletin 12, 343–348 (1984)
54. S.I. Kuchanov, S. Russo, Macromolecules 30, 4511–4519 (1997)
55. J. Guzman, E. Riande, J. Polym. Sci. Part A Polym. Chem. 25, 365–371 (1987)
56. A. Duda, R. Szymansky, S. Penczek, J. Macromol Sci, Part A Pure Appl. Chem. 20, 967–978 (1983)
57. N.T. McManus, A. Penlidis, M.A. Dube, Polymer 43, 1607–1614 (2002)
58. G.F. Cai, D.Y. Yan, Makromol. Chem. 186, 597–608 (1985)
59. D.Y. Yan, G.F. Cai, Makromol. Chem. 186, 2133–2144 (1985)
60. G.F. Cai, D.Y. Yan, J. Macromol. Sci, Part A Pure Appl. Chem. 24, 869–890 (1987)
61. A.M. vanHerk, A.L. German, Macromol. Theor. Simul. 7, 557–565 (1998)
62. J. Harwood, Angew. Chem. Internat. Ed. 4, 394–401 (1965)
63. X. Hu, D.Y. Yan, Makromol. Chem. Theo. Simul. 1, 161–171 (1992)
64. X. Hu, D.Y. Yan, A. Feng, Makromol. Chem. Theo. Simul. 1, 173–185 (1992)
65. K. Matyjaszewski, M.J. Ziegler, S.V. Arehart, D. Greszta, T. Pakula, J. Phys. Org. Chem. 13, 775–786 (2000)
66. U. Beginn, Colloid Polym. Sci. 286, 1465–1474 (2008)
67. S. Kuchanov, C. Kok, G. tenBrinke, Macromolecules 35, 7804–7814 (2002)
68. I. Goodman (ed), **Developments in Block Copolymers**, Applied Science Publishers, Barking, Essex, England, 1982
69. H. Tobita, S. Zhu, e-Polymers 25, 1–6 (2003)
70. S.I. Kuchanov, Adv. Pol. Sci. 152, 157–201 (2000)
71. M.R. Seger, G.E. Maciel, Anal. Chem. 76, 5734–5747 (2004)
72. H.N. Cheng, G.N. Babu, R.A. Newmark, J.C.W. Chien, Macromolecules 25, 6980–6987 (1992)
73. M.S. Montaudo, G. Montaudo, Macromolecules 25, 4264–4280 (1992)
74. H.J. Bae, T. Miyashita, M. Iino, M. Matsuda, Macromolecules 21, 26–30 (1988)
75. D. Braun, H. Elsasser, Macromol. Chem. Phys. 201, 2103–2107 (2000)
76. D. Braun, H. Elsasser, F. Hu, Eur. Polym. J. 37, 1779–1784 (2001)

77. G. Montaudo, R.P. Lattimer (eds), **Mass Spectrometry of Polymers, chapter 2**, CRC Press, Boca Raton, 2002
78. N.H. Cheng, Macromolecules 25, 2351–2358 (1992)
79. N.H. Cheng, Macromolecules 30, 4117–4125 (1997)
80. H.G. Barth, J.W. Mays, **Modern Methods of Polymer Characterization**, Wiley, New York, 1991
81. J. Koenig, **Spectroscopy of Polymers, 2nd edition**, Elsevier, NY, 1999
82. B.D. Coleman, T.G. Fox, J. Polym. Sci. Part A. Polym. Chem. 1, 3183–3197 (1963)
83. J.C. Randall, **Polymer Sequence Determination, the 13C NMR Method**, Academic Press, NY, 1977
84. M.F. Llauro, C. Monnet, F. Barbotin, V. Monteil, R. Spitz, C. Boisson, Macromolecules 34, 6304–6311 (2001)
85. A.M. Aerdts, J.W. deHaan, A.L. German, Macromolecules 26, 1965–1971 (1993)
86. L. Boggioni, F. Bertini, G. Zannoni, I. Tritto, P. Carbone, M. Ragazzi, D.R. Ferro, Macromolecules 36, 882–890 (2003)
87. A.L. Segre, M. Delfini, M. Paci, A.M. Raspolli-Galletti, R. Solaro, Macromolecules 18, 44–48 (1985)
88. A.L. Bailey, L.T. Kale, W.J. Tchir, J. Appl. Polm. Sci. 51 547–554 (1994)
89. V. Busico, R. Cipullo, A.L. Segre, G. Talarico, M. Vacatello, V. VanAxelCastelli, Macromolecules 34, 8412–8415 (2001)
90. G.H.J. VanDoremaele, A.L. German, N.K. De Vries, G.P.M. Van der Velden, Macromolecules 23, 4206–4215 (1990)
91. J.C. Randall, C.J. Ruff, Macromolecules 21, 3446–3454 (1988)
92. I. Kraemer, H. Pasch, H. Haendel, K. Albert, Macromol. Chem. Phys. 200, 1734–1744 (1999)
93. M.S. Montaudo, Macromolecules 34, 2792–2797 (2001)
94. M.S. Montaudo, Mass Spectrom. Rev. 21, 108–144 (2002)
95. G. Montaudo, E. Scamporrino, D. Vitalini, Macromolecules 24, 376–382 (1991)
96. M.S. Montaudo, Makromol. Chem. Theo. Simul. 2, 735–745 (1993)
97. G. Adamus, M. Kowalczuk, M.S. Montaudo, J. Polym. Sci. Part A Polym. Chem. 40, 2442–2448 (2002)
98. M.S. Montaudo, G. Montaudo, Macromolecules 32, 7015–7022 (1999)

Chapter 11
Predicting and Measuring the Sequence Distribution of Condensation Polymers

Maurizio S. Montaudo

Abstract The sequence of polycarbonates polyesters polyamides and other condensation copolymers is discussed. The theory predicts that the sequence is often radically different from addition polymers. One of the causes is the presence of a larger number of distinct monomers in the reaction vessel. Condensation copolymers from symmetric and asymmetric monomers involve a different number of parameters to be modeled, since asymmetric monomers require an additional parameter. A peculiar reaction is discussed, namely the synthesis of copolymers obtained by melt-mixing two different homopolymers. The sequence of copolymer fractions collected at the exit of a liquid chromatographic device is discussed as well. The measurement of condensation-polymer sequence distribution can be performed using nuclear magnetic resonance (NMR) and mass spectrometry (MS). Examples drawn from recent literature are discussed.

11.1 Introduction

Polymer science is a vast field and it is customary to classify polymerization reactions into two groups, namely chain-growth polymerization and step-growth polymerization [1–3]. In chain-growth polymerizations (also referred to as addition polymerizations), the chain grows in a well-defined manner, by adding one monomer at a time. The monomer always contains a C=C bond (triple bonds are very rare) and high masses are obtained at early reaction times. In step-growth polymerization (commonly referred to as condensation polymerization), a small molecule (water or methanol or chloridic acid) is often eliminated. The average molar mass increases slowly. High masses can be obtained only at high conversion [1–3]. Figure 11.1 reports two simple condensation polymerization reactions,

M.S. Montaudo (✉)
Institute of Chemistry and Technology of Polymers, CNR,
Via Paolo Gaifami, 18-95126 Catania, Italy
e-mail: mmontaudo@unict.it

R. Bruni (ed.), *Mathematical Approaches to Polymer Sequence Analysis and Related Problems*, DOI 10.1007/978-1-4419-6800-5_11,

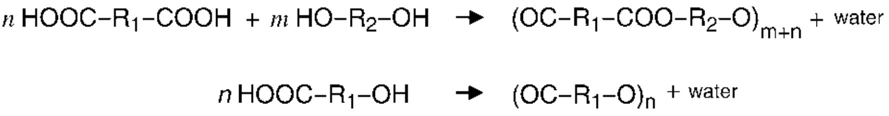

n HOOC-R1-COOH + m HO-R2-OH → (OC-R1-COO-R2-O)m+n + water
n HOOC-R1-OH → (OC-R1-O)n + water

Fig. 11.1 Two simple condensation polymerization reactions

namely the synthesis of a polyester using a diacid (a molecule with two COOH groups) and a diol (a molecule with two OH groups) and the synthesis of a polyester using an hydroxy-alcohol: the symbols R1 and R2 that appear in the scheme are functional groups. One of the simplest cases is when both are methylenes (R1=$-CH_2-$ and R2=$-CH_2-$). Polycarbonates are condensation polymers. The polymer called Nylon 6 belongs to the group of Polyamides. It is obtained industrially by a "ring opening" synthetic route [1–3]. In the following, we shall describe a series of different polymerization reaction conditions and, for each different reaction, we will derive a prediction for the sequence of the reaction products. This will be followed by a section on experimental methods to measure the sequence.

11.2 Sequence Prediction

In order to define the quantities of interest, it is necessary to recall some concepts from the previous chapter. A polymerization reaction is a chemical reaction in which reactants are transformed into reaction products. The reactants are called the feed. The feed contains monomers at a concentration of z_{tot}. In binary copolymers the feed contains A and B at a concentration of z_{A} and z_{B} and thus the sum of the molar ratio of A and B units in the feed, $f_{\mathrm{A}} + f_{\mathrm{B}}$, equals one. In a similar manner, the sum of the molar ratio of A and B units in the reaction products (i.e., in copolymer chains), $c_{\mathrm{A}} + c_{\mathrm{B}}$, equals one indeed. A sequence made of three repeat units is called triad. Tetrads, pentads, hexads, and heptads are sequences made of four, five, six, and seven repeat units, respectively. Some polymers possess a high oligomeric content. In oligomers, the chain is short (by definition). The dimer contains two repeat units, whereas the trimer, tetramer and the pentamer contain three, four, and five repeat units, respectively. The Molar Mass Distribution (MMD) measures the abundances of chains of length s. MMD averages $\bar{\prod}_n$ and $\bar{\prod}_w$ are important. The Schulz–Zimm MMD function is given by the product of a power law and a decreasing exponential:

$$\mathrm{MMD}(s) = a_{\mathrm{nofa}}(s)^{\alpha}\exp(-s/\theta), \qquad (11.1)$$

where α and θ are two adjustable parameters and a_{nofa} is a suitable normalization factor. When $\alpha = 0$, the distribution is a decreasing exponential and thus $\bar{\prod}_w$ doubles $\bar{\prod}_n$:

$$\bar{\prod}_w = 2\bar{\prod}_n \qquad (11.2)$$

Let $W(m,n)$ denote the weight of the copolymer chains of the type $A_m B_n$ (which are made of m repeat units of type A and n repeat units of type B). $W(m,n)$ is given by

$$W(m,n) = \sum\nolimits_{\mathrm{XXXX} \in \Omega} W_{\mathrm{XXXX}}, \tag{11.3}$$

where W_{XXXX} is the weight of copolymer chains having sequence XXXX and where the summation spans over the set, Ω, of sequences XXXX which are made of m repeat units of type A and n repeat units of type B. One can introduce y, the deviation from the average composition, defined as $y = (c_{\mathrm{A}} - m/s)$. In this way, $W(m,n)$ becomes $W_s(y)$, referred to as the bivariate distribution. Integrating over sizes the bivariate distribution, one obtains CODIHI, the compositional distribution histogram.

$$\mathrm{CODIHI} = \int W_s(y)\mathrm{d}s' \tag{11.4}$$

In predicting the sequence of condensation copolymers, the symmetry of the monomer is of atmost importance. After careful inspection of reactions in Fig. 11.1, one can realize that the upper reaction involves monomers of the type AA or BB (which possess a plane of symmetry). Actually, the term *monomer* may have different meanings. In the polyesters depicted in the upper and lower parts of Fig. 11.1, the monomer and the repeat unit are highly dissimilar and highly similar, respectively. In the former, the monomer is only a small part of the repeat unit. In the latter, the monomer and the repeat unit are identical (apart a water molecule). Also in poly(lactone)s and poly(lactam)s, the repeat unit and the monomer are similar. A review by Kuchanov et al. [4] gives a comprehensive view of all theories on condensation polymerization and co-polymerization. It even covers the initial efforts (such as the theory proposed by Flory between the two world-wars). We will discuss most of these theories too. Theories that appeared in Russian journals will be omitted for brevity. We will begin with the relatively simple case in which the monomer is symmetric, i.e., it is of the type AA. Thereafter, we will discuss asymmetric monomers, cyclic dimers, and melt-mixing (reactive blending).

11.3 Condensation Copolymerization Using Symmetric Monomers

A review by Marechal and Fradet deals with condensation copolymerization [5] and it possesses a section with critical comments on virtually all the theories that appeared on the subject, including the theory proposed in 1958 by Case, the Monte

Carlo computations performed by Chaumont, Gnagou et al. and the theory proposed by Lopez-Serrano, Macosko et al. (with the derivation of explicit formulas for $< n_{\mathrm{A}} >$, $< n_{\mathrm{B}} >$ the number-average sequence length).

In poly(ethylene-terephthalate), poly(butylene-adipate), and other polymers, the monomers are paired and thus the repeat unit is made of two monomers, namely ethylene glycol and terephthalic acid, butanediol and adipic acid, more in general, a diol and a diacid. The development of a mathematical model which uses two paired monomers as a single entity is certainly possible (it is identical to the models developed for addition copolymers). However, it cannot account for a very well-known phenomenon, namely stoichiometric unbalance (in fact, paired monomers are necessarily stoichiometrically balanced). To get high-molecular copolycondensates, it is necessary to conduct the reaction up to nearly complete conversion. In that case, the copolymer average composition, cA will be virtually the same as that of the initial monomer mixture, and thus the problem of its determination ceases to be actual. Thus, the Bernuolli model holds. It predicts that the abundance of an oligomer is equal to the product of the molar fractions of the monomers. For instance, the abundance of oligomer AABBAACDDCAAB is given by the following product:

$$I_{\mathrm{AABBAAAAAB}} = c_{\mathrm{A}}c_{\mathrm{A}}c_{\mathrm{B}}c_{\mathrm{B}}c_{\mathrm{A}}c_{\mathrm{A}}c_{\mathrm{A}}c_{\mathrm{A}}c_{\mathrm{A}}c_{\mathrm{A}}c_{\mathrm{A}}c_{\mathrm{B}} \tag{11.5}$$

The quantities of interest are the molar fraction, $I(A_m B_n)$, of the oligomer $A_m B_n$, the molar fraction, $I(A_m B_n C_p)$, of oligomer $A_m B_n C_p$, and the molar fraction, $I(A_m B_n C_p D_q)$, of oligomer $A_m B_n C_p D_q$. They are obtained by combinatorial calculus, summing all contributions:

$$I(A_m B_n) = \tfrac{(m+n)!}{m!\ n!}(c_{\mathrm{A}})^m (c_{\mathrm{B}})^n \tag{11.6}$$

$$I(A_m B_n C_p) = \tfrac{(m+n+p)!}{m!\ n!\ p!}(c_{\mathrm{A}})^m (c_{\mathrm{B}})^n (c_{\mathrm{C}})^p \tag{11.7}$$

$$I(A_m B_n C_p D_q) = \tfrac{(m+n+p+q)!}{m!\ n!\ p!\ q!}(c_{\mathrm{A}})^m (c_{\mathrm{B}})^n (c_{\mathrm{C}})^p (c_{\mathrm{D}})^q \tag{11.8}$$

The above equations are often referred to as the Newton and the Liebnitz formulas, respectively. Figure 11.2 reports the theoretical mass spectra for a copolymer, a terpolymer, and a tetrapolymer, respectively. The masses correspond to units of butyleneadipate butylenesuccinate, butylenesebacate, and butyleneterephthalate [6]. In this way, the comparison with experiment is simple. It can be seen that the number of peaks grows as the size of the oligomer grows. In the case of binary copolymers, the number of peaks [7] grows as $(s + 1)$. In the case of tetrapolymers, the spectrum becomes crowded at the trimer level (see figure).

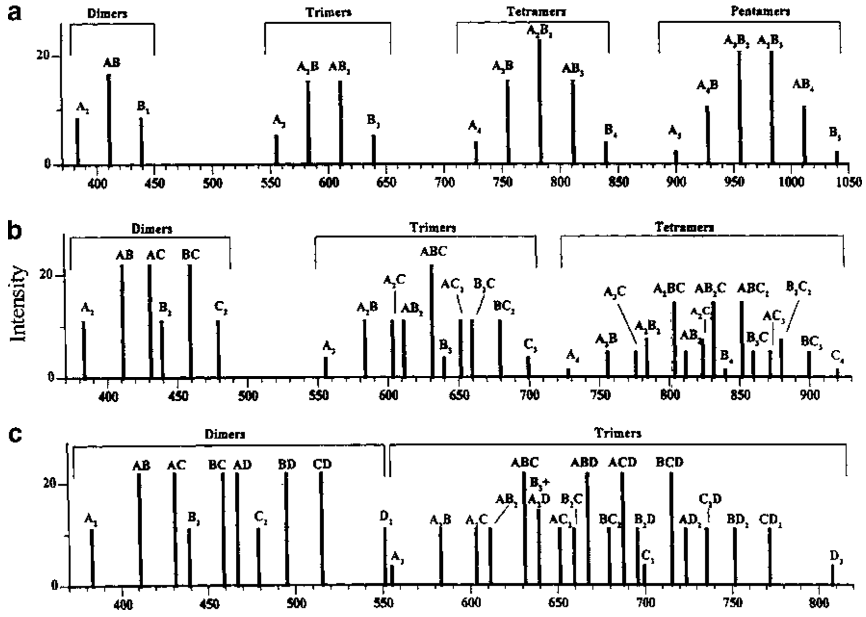


Fig. 11.2 Theoretical mass spectra for a copolymer (**a**), a terpolymer (**b**), and a tetrapolymer (**c**) respectively. Reprinted from [6] with permission from the copyright holder ACS 1998

11.4 Condensation Copolymerization Using Asymmetric Monomers

Figure 11.3 reports the chemical structure of an asymmetric monomer and, more specifically, an asymmetric diol. When it is splitted in the middle, it does not leave two identical parts. Lactones and lactams are other two examples of asymmetric monomers. In this case, the monomer is a ring. Lactones are used to produce poly(epslon-caprolactone) and other poly(lactone)s. Lactams are even more important, since they are used to produce Nylon6 and other nylons. Industrially, some chemicals are used more than others. Important diisocyanates are methylene bis(para phenyl isocyanate) (MDI for brief) and 2,4-toluene diisocyanate (TDI for brief). An important chain extender is PTMO1000, which is made of poly(tetramethylene oxide) chains of 1,000 g mol^{-1}. Speckhard et al. [8, 9] developed a Monte Carlo algorithm which predicts the sequence of condensation copolymers obtained from symmetric and asymmetric monomers and applied it to copolyurethanes obtained reacting three monomers, namely a diisocyanate (MDI or TDI), a long glycol (such as hydroxy-terminated polybutadiene), and a chain extender (PTMO1000). The method developed by Zetterlund et al. [10, 11] also relies on a Monte Carlo algorithm. Ray and Jacobsen applied the method of moments to solve the differential equations [12]. These numerical methods are certainly useful, but our interest is in exactly soluble models. Vasnev and Kuchanov [13] proposed

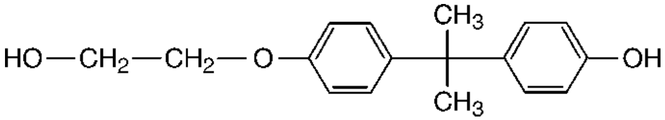


Fig. 11.3 An asymmetric diol

the *raznozvennost* theory, which is characterized by a microheterogeneity index, K_{M}, and they calculated K_{M} values as a function of the reactivity of the functional groups present in the reaction mixture. They noted that the abundance of dyads I_{XX} and triads I_{XXX} are related, and they reported pertinent relationships. Schematically

$$I_{\mathrm{XX}} = \phi_{\mathrm{trim}} \left(\sum_{\mathrm{X}} I_{\mathrm{XXX}} \right), \tag{11.9}$$

where ϕ_{trim} is characterized by the presence of the product of two Kronecker deltas. They also introduced an orientation coefficient, which is claimed to be useful in the case of asymmetric diols. Unfortunately, they adopted a cumbersome notation, with arrows to indicate the orientation of the diol. As a result, it is difficult for the reader to figure out if the copolymer sequence is Bernoullian or first-order markovian.

Another theory which can deal with asymmetric diols (such as the one in Fig. 11.3), with lactones, lactams and other asymmetric monomers, is the half-monomer theory [14]. The copolymer sequence is second-order markovian in four monomers and constrains of the type A cannot follow A, B cannot follow B, etc. are adopted. In this model, the abundance of long sequences, XXWYZ, is obtained from shorter oligomers sequences using the standard second-order markovian recurrence formula:

$$I_{\mathrm{XXWYZ}} = I_{\mathrm{XXWY}} Q_{\mathrm{WYZ}}, \tag{11.10}$$

where Q_{WYZ} are **Q**-matrix element. The above equations were implemented in a computer program called Nuclear magnetic resonance Analysis of Condensation copolymers by COnstrained Second-order markoffian (NACCOS).

11.5 Condensation Copolymerization Using Cyclic Dimers

Since cyclic dimers are asymmetric monomers, in line of principle they should be described by the theories in the previous section. In practice, however, specific theories are needed [15,16]. Certainly, the most important cyclic dimer is lactide, i.e., the lactic acid cyclic dimer. A second monomer is needed in order to obtain a copolymer. The latter can be a carbonate, an amide, an ester from a diacid and a diol, a lactone, or another cyclic dimer. The resulting copolymer sequence is quite complex for the following reason. A genuine model for this reaction necessarily makes use of dimers for the description of the copolymer. The model needs dimer-to-monomer transformation formulas which allow sequence description in terms of monomers.

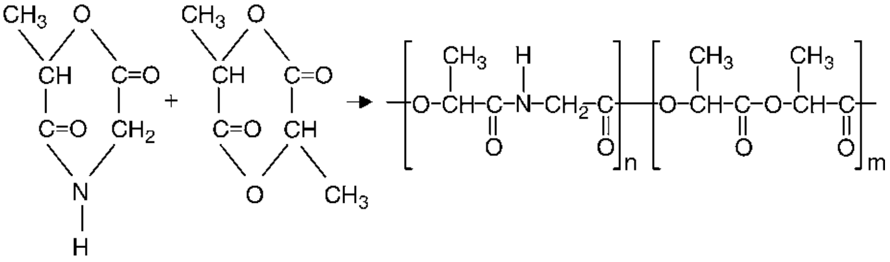


Fig. 11.4 The synthesis of a copoly(ester-amide) starting from two cyclic monomers

The latter formulas are not easy to derive. Figure 11.4 reports a copolymerization reaction involving the lactic acid cyclic dimer and another cyclic dimer.

11.6 Bacterial Synthesis of Condensation Copolymers

Bacteria such as *Pseudomonas olevarans*, *Pseudomonas putida*, *Alcaligenes euthrophus*, *Rodospirillum rubrum*, and *Ralstonia eutropha* are able to produce polymers and copolymers, mainly polyesters and copolyesters [17]. Bacterial synthesis is the result of many chemical reactions. No wonder, therefore, if the sequence of the resulting copolymer is unknown in the sense that nobody has ever proposed a theory that predicts the cited sequence.

It is reasonable to assume that the copolymer is a mixture of copolymers. As discussed in detail previously, mixtures of two copolymers display sequences due to both components of the mixture. The molar fraction, $I(A_m B_n)$, of the oligomer $A_m B_n$ is given by:

$$\begin{aligned} I[A_m B_n] = \frac{(m+n)!}{m!\,n!}\,(&E_{\text{mix}}(e_{\text{A}})^m(1-e_{\text{A}})^n \\ &+ (1-E_{\text{mix}})(d_{\text{A}})^m(1-d_{\text{A}})^n)\,, \end{aligned} \tag{11.11}$$

where e_{A} and d_{A} are the compositions (the molar fraction of A units) of the first and the second copolymer, and E_{mix} is the molar fraction of the first copolymer in the mixture. From the above compact formula, the explicit expressions for each oligomer can be derived. The most useful ones are those from dimers to hexamers [18].

11.7 Condensation Copolymerization by Melt-Mixing

The topic of exchange reactions in condensation polymers is vast and reviews covering the entire topic or part of it have been published [19–22]. Binary polymer melts are obtained by heating binary polymer mixtures above the melting temperature, T_{m}, which is often above 200°C. In some binary polymer melts (e.g., polyester–polyester, polyester–polyamide, polyester–polycarbonate, and polyamide–polyamide) exchange reactions occur. Depending on the chemical groups found along the macromolecular chain, various types of exchange reactions in polymers occur, namely ester–ester exchange [23], ester-carbonate exchange [24], carbonate–carbonate exchange [25], amide–ester exchange [26], amide–carbonate exchange [27], and amide–amide exchange [28]. When the blend is heated, the two homopolymers can react together to form a copolymer. The above process can be described as follows:

$$\mathrm{AAAAAAAA} + \mathrm{BBBBBBBB} \rightarrow \mathrm{AAABBBBB} + \mathrm{AAAAABBB} \tag{11.12}$$

The copolymer has initially long AAAAAA and BBBBB blocks but, as the reaction goes on, the length of the blocks decreases and it becomes random. This second process can be described as follows

$$\mathrm{AAAAABBBBB} \rightarrow \mathrm{AAABBBBBAA} \tag{11.13}$$

Most condensation polymers blends are incompatible at room temperature. In fact, a phase-separation quickly occurs in the blend, with A-rich and B-rich phases. The dimension of these two phases can be characterized by a "phase average diameter" (PHAD). Above the melting temperature, the segmental mobility is high and the blend is compatible. However, once the blend is cooled, two phases are generated and PHAD increases slowly with time, which implies that the blend is unstable. Phase-separation must be avoided, otherwise the blend's characteristics are utterly depleted. In order to overcome (at least in part) this drawback, one can add a compatibilizer. The latter are usually interfacially active copolymers (for instance, block or graft copolymers) which possess at least two different types of segments, and each type is capable of specific interactions with one of the blend's components. The action of a compatibilizer can be depicted as follows. There are two phases (I and II) separated by a phase boundary. The compatibilizer joins the two phases, since it lies in partly in phase I and partly in phase II. Thanks to the compatibilizer, the overall energy of the blend is reduced and this has benefits on the mechanical properties. In some cases, one uses macromolecular compatibilizer that are totally different from the blend's component. In the case of condensation copolymers, however, this strategy has a most evident shortcoming. In fact, condensation copolymers are often used as engineering thermoplastics and high compatibilizer concentrations must be avoided, otherwise the presence of additional repeat units may cause unwanted effects. Vice versa, it is quite apparent that the best compatibilizer for a blend of $\mathrm{A}+\mathrm{B}$ is an AAAAABBBB block copolymer. In fact, the system will evolve in such a way to minimize its overall energy, and it is intuitive that AAAA blocks will lie in phase I and BBBB blocks will lie in phase II. Furthermore, the best compatibilizers are those that have a relatively high molar mass so that the average length of AAAA blocks is high enough to form entaglements in the phase I. It is apparent that the exchange reaction depicted in (11.13) yields a block copolymer which is ideally suited to act as compatibilizer. The formation of copolymers in blends by exchange reactions has been studied theoretically [29, 30]. A theory was developed which predicts how the sequence varies as the abundance of the reaction products increases. As a matter of fact, the sequence is described by a single formula, valid for all reaction times:

$$N_w(\tau) = \frac{2N_{\text{ble}}}{1 - \mathrm{e}^{-\tau}(1 + G_{\text{gal}})^{-1}}, \tag{11.14}$$

where N_{ble} is the number-average polymerization degree of the blend (directly proportional to $\bar{\prod}_n$), τ is the number of cleavages per chain of the number-average polymerization degree, and G_{gal} is a function of τ. This is amazing since, at later stages, the copolymer sequence can be described by a statistical formula (a first-order markovian), but in the initial stages, the sequence cannot be described by a statistical formula (B units are not found in long AAAA blocks). Unfortunately, the cited theory does not predict two important quantities, namely the BIVA (defined by (11.3)) and the CODIHI (defined by (11.4)). In order to cope up with these limitations, a new theory has been proposed [31], in which CODIHI is obtained integrating the BIVA over chain sizes. The resulting formula is:

$$\text{CODIHI} = Z1(t) + Z2(t) + Z3(t) \tag{11.15}$$

The above equation states that the sample is a complex mixture made of three components or simply "parts," referred to as Z1, Z2, Z3, where Z1 and Z2 are the parts for unreacted homopolymers (A and B), whereas Z3 is the part for the copolymer. The predictions of the above equation are masked by its formal simplicity. In order to unviel them, let us consider the case when a liquid chromatography (LC) apparatus is used. If the crude mixture obtained by reactive blending of two polymers is injected in the apparatus, the volume at which a macromolecule is eluted, V_e, will depend on chain size and on the molar ratio, r_{A}, associated with the chain (r_{A} equals to the ratio m/s, where s is the chain size and m are the number of A units in the chain). It has been reported that, for some copolymer systems, conditions can be found in which V_e is independent from molar mass. The cited systems possess a peculiarity, namely the two repeat units A and B possess very different structures. A mixture of PC and Nylon 6 was considered. They are structurally different since the first one possesses two aromatic rings (Fig. 11.5), whereas the second one is fully aliphatic. The algorithms of the new theory were implemented in a computer program called TRIPLEBIV and a calculation was performed assuming that LC separates in a size-independent manner. Figure 11.6 reports the simulated LC chromatogram for a copolymer PC-Nylon 6 obtained by melt-mixing at two different times. For sake of simplicity, it was assumed that V_e follows a linear law of

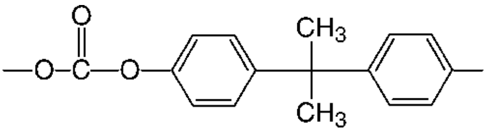


Fig. 11.5 The repeat unit of poly(bisphenol-A-carbonate)

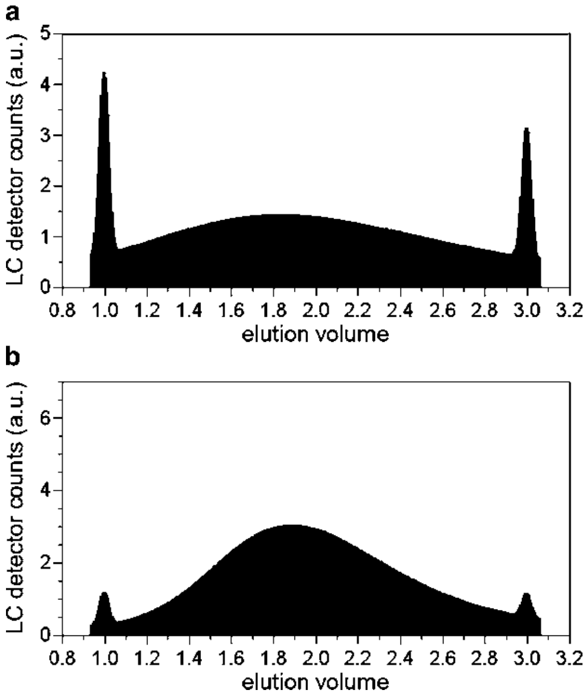


Fig. 11.6 Simulated LC chromatogram for a copolymer PC- Nylon 6 obtained by melt-mixing at $t = 3\,\text{min}$ (**a**), At $t = 8\,\text{min}$ (**b**)

the type $V_e = g_0 + g_1 r_\text{A}$, with $g_0 = 1\,\text{mL}$ and $g_1 = 2\,\text{mL}$, and that the resolving power is almost infinite. As a result, the two homopolymers elute almost exactly at $V_e = 1\,\text{mL}$ and $V_e = 3\,\text{mL}$, with very small spreads around these values. On the other hand, the peak due to copolymer chains is centered around $V_e = 2\,\text{mL}$ and it is very broad. Actually, it is so broad that it superposes with Z1 and Z2. This constitutes a small obstacle that can be easily circumvented using a deconvolution process to estimate Z1 and Z2. The simulation clearly indicates that homopolymer chains disappear at early stages of the exchange reaction.

11.8 The Sequence of Copolymer Fractions

Copolymer fractions are usually collected at the exit of a liquid chromatographic device. There are several techniques that yield copolymer fractions. For instance, precipitation fractionation, extraction fractionation (using a simple SOXHLET extractor), gradient elution fractionation, field flow fractionation (FFF), supercritical CO^2 fractionation, and temperature-rise-elution fractionation (TREF) [32]. The composition (molar fraction of A units) of the mth fraction and of the unfractionated copolymer will be denoted by $c_{\mathrm{A}}(m)$ and c_{A}, respectively. The difference $|c_{\mathrm{A}}(m) - c_{\mathrm{A}}|$ can be large. In order to measure this effect quantitatively, one may introduce a fractionation-efficiency parameter, D_{FRA}, defined as:

$$D_{\mathrm{FRA}} = \sigma^2_{\mathrm{EXP}}/\sigma^2_{\mathrm{THEO}}, \qquad (11.16)$$

where σ^2_{EXP} and σ^2_{THEO} are the experimentally obtained and true variance of chemical composition distribution, respectively. Topchiev and coworkers modeled copolymer fractions starting from Flory–Huggins theory, and the prediction is that their sequence is not identical to the sequence of unfractionated copolymer [33]. Kratochvil and coworkers [34] used (11.16) and noted that butanone–methanol is very powerful for separating copolymers. Equation (11.16) can be used also in the case of block copolymers [35]. Eersels et al. [36] used gradient elution fractionation to separate by composition a condensation copolymer obtained by melt-mixing of PA 4,6 (a fully aliphatic polyamide) and PA 6I (an aromatic polyamide obtained reacting diaminohexane with isophthalic acid). They did not measure the separation efficiency. Nevertheless, it was high, since the elution volumes for the two homopolymers ($V_e = 40$ mL and $V_e = 70$ mL, respectively) are quite different. Recently, a new technique called liquid chromatography at the critical condition (LCCC) has been developed. It works at the elution-adsorption transition, and, thus it can separate copolymers characterized by different compositions with a virtually infinite efficiency ($D_{FRA} = 1$). LCCC separation has been used to characterize a copolyesters made of adipinic acid, neopentylglycol and hexanediol [37], and a poly(L-lactide)-block-poly(ethylene oxide)-blockpoly(L-lactide) (PLLA-b-PEO-b-PLLA) triblock copolymer [38].

11.9 The Sequence of SEC Fractions

Copolymers can be separated by size exclusion chromatography (SEC). The reader may be induced to think that the separation mechanism acts exclusively on sizes. This is false. It acts also on compositions. The sequence of SEC copolymer fractions can be modeled [39]. The molar fraction, $I(A_m B_n)$, of the oligomer $A_m B_n$ is given by:

$$I[A_m B_n] = g_1 \exp(y + J_{\mathrm{seq}})Q, \qquad (11.17)$$

where g_1 is a suitable normalization factor, J_{seq} (the sequence distribution) is given by $J_{\mathrm{seq}} = [m/s - d]/h$, with $d = d_0 + d_1 s + d_2 s^2$, where d_0, d_1, d_2 are parameters which describe the sample's compositional heterogeneity, where h is the width of the CODIHI, y is an MMD which obeys the condition $y = y_1 \log(s) - s/y_2$ where y_1 and y_2 are MMD-shaping parameters given by $y_1 = (2\bar{\prod}_n - \bar{\prod}_w)/(\bar{\prod}_w - \bar{\prod}_n)$, $y_2 = \mu/(\bar{\prod}_w - \bar{\prod}_n)$, and $\bar{\prod}_n$, $\bar{\prod}_w$ are the number- and weight-average molar masses, respectively, of the unfractionated copolymer. Q is a function which describes the effect of the separation process. At the exit of the chromatographic device, SEC fraction x contains chains $A_m B_n$ of a given size, and it does not contain chains $A_m B_n$ of other sizes. Clearly, Q is always equal to zero, except in a small range of elution volumes between V_{ini} and V_{fin}. When the separation process is extremely efficient, one has:

$$Q = H(V - V_{\text{ini}})H(V_{\text{fin}} - V), \tag{11.18}$$

where the function H is equal to zero when its argument is negative and it is equal to 1 when its argument is positive. The sequence in (11.17) is fully new and quite different from the sequence of unfractionated copolymer.

11.10 Sequence Descriptors

As stated in the chapter that deals with addition copolymers, sequence descriptors are used to describe the sequence in a compact manner. They are universal, in the sense that they can be used to describe any kind of copolymer, independent of whether it is a condensation copolymer or an addition copolymer. Despite this universality, some authors proposed sequence descriptors specifically suited for condensation copolymers. The most important ones are the Yamadera–Murano degree of randomness [40], the four-components-condensate degree of randomness [41], the Lenz–Jin–Feichtinger randomness number [42], the preference factor proposed by Ou [43], the Kricheldorf–Saunders heterodyad/homodyad ratio [44], Kasperczyk's BLR/BLO ratio [45] (where BLR and the BLO are the random and the observed block lengths, respectively), and the sequence-order parameter introduced by Lyerla and coworkers [46]. Tessier and Fradet [47] discussed all the above sequence descriptors. They also derived formulas that relate these "new" sequence descriptors and the conditional probabilities (the P-matrix elements). These formulas are extremely useful. Some comments are in place. The number-average block lengths (denoted by $< n_{\text{A}} >$ and $< n_{\text{B}} >$) are powerful sequence descriptors. A sequence can be described in three ways, namely in a extended manner, in a semi-compact manner, and in a compact manner. In order to avoid confusion, the first and second options are preferable.

11.11 Sequence Measurement

The measurement of the sequence distribution of addition polymers can be performed using various techniques [1–3]. In the previous sections, we already showed how LC can be applied to the analysis of a mixture of PC and Nylon6. Two other methods will be discussed here, namely nuclear magnetic resonance (NMR) and mass spectrometry (MS).

11.12 Nuclear Magnetic Resonance

NMR can be applied to the analysis of condensation polymers. For instance, a copolymer with units of etherketone and ethersulfone [48] was sequenced by NMR. The sample was treated as a mixture of chains and, more specifically, a binary mixture characterized by a mole fraction $(1-\chi)$ of exactly alternating chains and a mole fraction (χ) of bernoullian chains. Condensation terpolymers can be sequenced by NMR too. Su and Shih analysed by NMR a ternary mixture made of a poly(ethylene naphthalate)/poly(trimethylene terephthalate) copolymer and a poly(ether imide) [49]. Kricheldorf and Hull sequenced by NMR a terpolyamide [50]. However, NMR turns out to be less powerful for condensation polymers than for addition polymers. This is due to the fact that condensation polymers possess bulky repeat units. Unfortunately, NMR can hardly discern beyond the triad level. When this statement is translated in the "repeat unit" language, it becomes: NMR can hardly discern beyond 1.5 repeat units. This implies that the penultimate model, the pen-penultimate model, and other models with a high number of degree of freedom cannot be used.

11.13 Mass Spectrometry

In the previous sections, we briefly introduced MS and the theoretical mass spectra for a copolymer, a terpolymer, and a tetrapolymer were shown. We also considered condensation copolymerization with cyclic dimers from a theoretical point of view. MS can be applied to the analysis of the above copolymers. In particular co-polydepsipeptides are currently under consideration as biodegradable materials. Ring-opening copolymerization of lactide and 6-methyl-2,5-morpholinedione (the cyclic glycine-lactic acid dimer) affords a series of co-polydepsipeptides corresponding to Fig. 11.4. It is apparent that the synthetic route yields a quasi-random copolymer with a peculiar sequence distribution, since it deviates from the standard form in which the two repeat units (glycine and lactic acid residues) are found at random along the copolymer chain. The deviation is due to chains having sequences containing two consecutive glycine units, GG, which are not produced. This deviation is small and is therefore difficult to detect. Figure 11.7 reports the MALDI-TOF spectrum of a copolymer containing units of glycine (G) and lactic acid (L) [51]. The peaks in the mass spectrum were assigned to oligomers bearing the same end groups, namely:

$$\mathrm{H}-(\mathrm{O}-\mathrm{CH}(\mathrm{CH}_3)-\mathrm{CO})_m-(\mathrm{NH}-\mathrm{CH}_2-\mathrm{CO})_n-\mathrm{OH} \qquad (11.19)$$

The peak at $m/z = 401$ is due to the lactic pentamer (L_5). The lactic hexamer falls at $m/z = 473$, and it displays a more than double intensity. An intense peak in the mass spectrum is at $m/z = 515$ and it corresponds to an oligomer with five lactic acid residues and two glycine residues (L_5G_2L). The peak at $m/z = 530$ is slightly less intense and it is due to L_6G_1. Pentamers, heptamers, and other

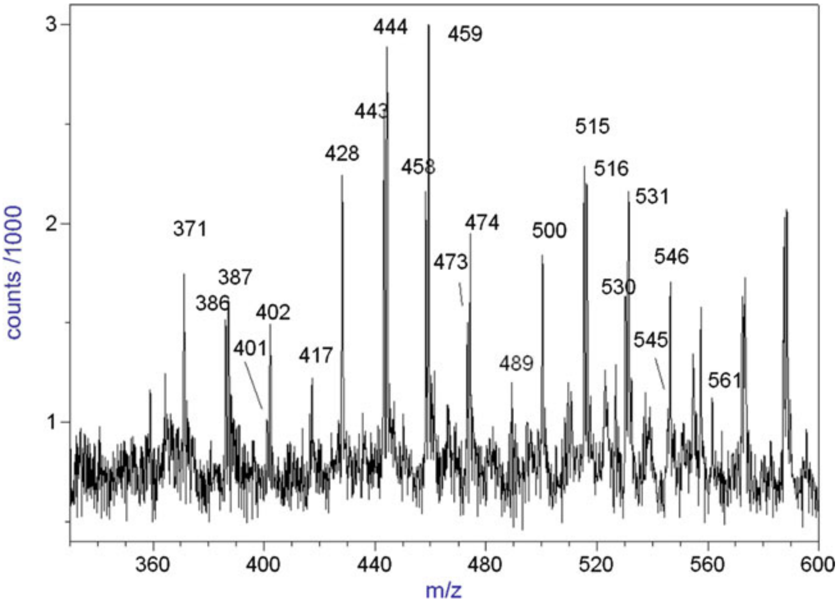


Fig. 11.7 MALDI-TOF spectrum of a copolymer containing units of glycine (G) and lactic-acid (L). Reprinted from [51] with permission from the copyright holder Wiley 1999

odd-membered oligomers are present in the sample. The polymerization reaction (Fig. 11.4) should produce exclusively even-membered oligomers. Possible explanations for the presence of odd oligomers may be the hydrolysis of the polymer chain or an ester–ester exchange reaction occurring during polymerization. This phenomenon has already been observed in poly(lactic) produced by ring opening of the dilactide. The fact that the intensity of MS peak due to oligomers which possess many lactic residues implies that the average molar fraction of L units, c_{L}, is larger than 0.50. MS was used to measure copolymer's sequence. More specifically, sequence information was extracted from mass spectral peak intensities using a computer program called mass Analysis of COpolymers (MACO). The output was $c_{\mathrm{L}} = 0.77$, which compares well with the composition determined by NMR, namely $c_{\mathrm{L}} = 0.76$. Even more important, the MACO method was also able to detect subtle deviations within a sequence in which lactic and glycine residues alternate at random along the chain. In the previous sections, we pointed out that bacteria can produce polymers. A series of bacterial copolyester, in which β-hydroxybutyrate (HB) and β-hydroxyapatite (HV) units are found along the chain, were analysed. In principle, using (11.11), MS is able to discriminate a genuine HB–HV copolymer from a mixture of two copolymers. Some experiments were conducted to see whether this is true. Three HB–HV copolyester samples, denoted as S6, S7, S3 were considered. Sample S6 is an equimolar mixture of homopolymer HB plus HB–HV copolymer (composition 72/28). Thus, the overall molar fraction of HB units, c_{HB}, is 0.86. Sample S7 is an equimolar mixture of HB–HV copolymer (composition

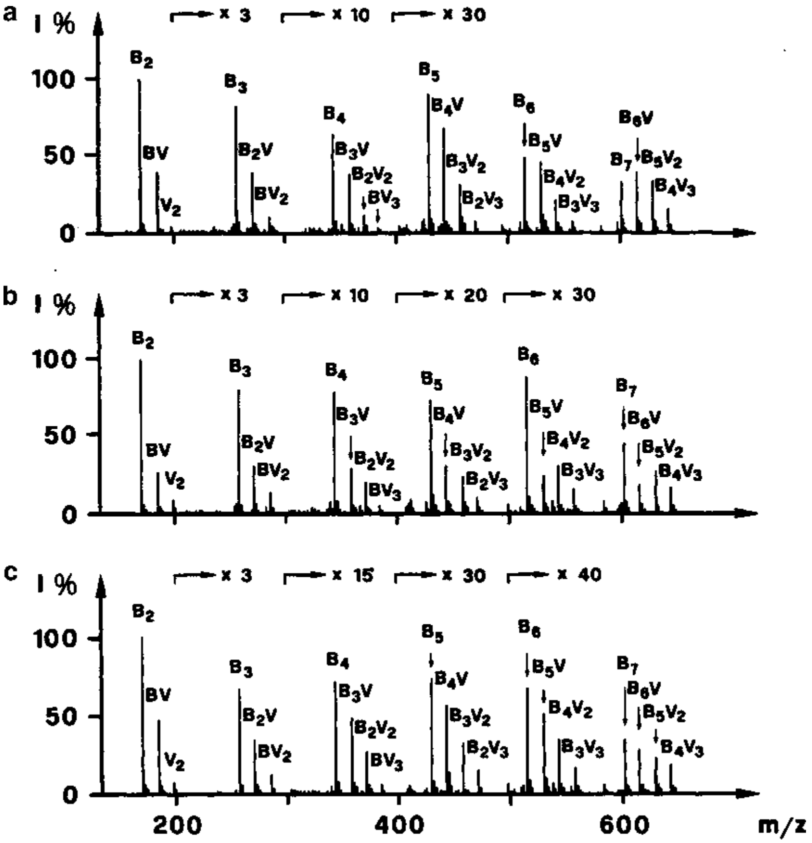


Fig. 11.8 FAB spectra of the three HB-HV copolyesters denoted as S6 (**a**), S7 (**b**), S3 (**c**). Reprinted from [52] with permission from the copyright holder ACS 1991

of 94/6) plus HB–HV copolymer (composition 72/28). Thus, $c_{\text{HB}} = 0.83$. Sample S3 is not a mixture, it is a genuine HB–HV copolymer. The copolymer composition is very similar to the preceeding ones, $c_{\text{HB}} = 0.83$. Figure 11.8 reports FAB spectra [52] of the three HB–HV copolyester S6, S7, S3. The MS intensities are different indeed. From the analysis of MS data, it was concluded that MS is able to discriminate a genuine HB–HV copolymer from a mixture. Bacterial copolyester sequencing by MS is now performed routinely [53]. Poly(ethylene -terephthalate) (PET for brief) is important since it is used in plastic bottles and other items. The sequence distribution and composition of the copolymers generated by melt mixing of poly(ethylene terephthalate)/poly(ethylene adipate) blends (PET–PEA) were determined by analysis of the FAB mass spectra of the oligomers present in the crude blends or else formed by appropriate partial degradation (hydrolysis or aminolysis) of the mixtures. Figure 11.9 shows the positive ion FAB spectra of the oligomers present in the crude PEA–PET blend melt-mixed at 290°C for increasing times

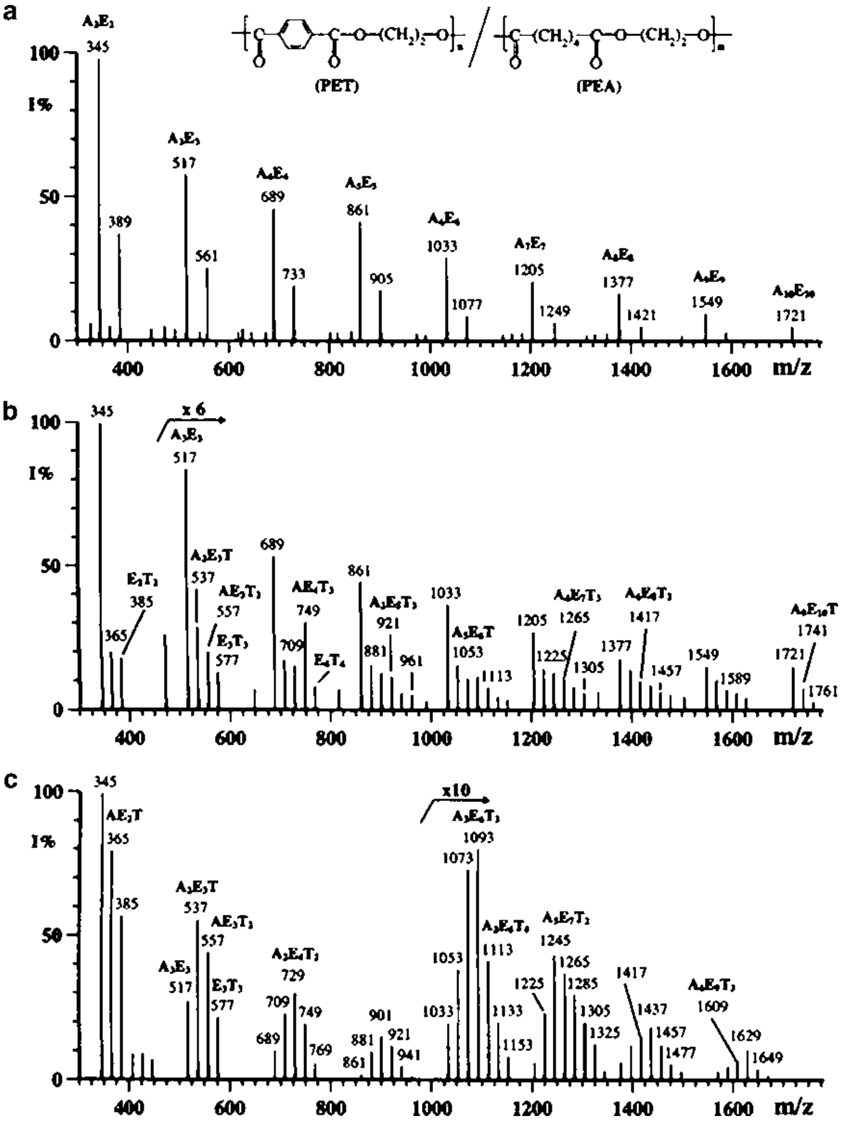


Fig. 11.9 Positive ion FAB spectra of the oligomers present in the crude PEA-PET blend melt-mixed at 290°C for 0 min (**a**), for 20 min (**b**), for 270 min (**c**). Reprinted from [54] with permission from the copyright holder ACS 1992

(0, 20, and 270 min, respectively) [54]. The FAB spectrum of the initial mixture (M0, Fig. 11.9a) is quite similar to that of pure PEA and does not show peaks corresponding to PET oligomers. In fact, the PET sample used for the blend had been completely freed from all the low molar mass oligomers before mixing it with the PEA. The mass spectra of the samples obtained after 20 min (M20, Fig. 11.9b) and after 270 min mixing (M270, Fig. 11.9c) show a series of peaks corresponding to protonated molar ions of cyclic homo- and co-oligomers. Peaks due to open-chain oligomers, not observed in the positive ion FAB spectra, were detected in the negative ion FAB mode. The MS peak intensities of the PET–PEA samples reacted from 10 up to 270 min were used to estimate the copolymer composition, the extent of exchange, and the number-average block lengths by the chain statistics modeling of the mass spectra of copolymers [54]. The extent of exchange, EE, is given by the sum of the P-matrix elements which belong to the principal diagonal, $P_{\mathrm{AB}} + P_{\mathrm{BA}}$. It coincides with the quantity Z3 in (11.15):

$$\mathrm{EE} = Z3 \tag{11.20}$$

This equation allows to correlate two different types of experiments (LC against MS). Fully aliphatic copolyesters based on ethylene adipate or butylene adipate are sold under the trade names Bionelle and Bionolle [55]. A copolymer containing units of butylene adipate (A) and units of butylene sebacate (B) was obtained using a synthetic route that makes use of butandiol and the dimethyl esters of adipic and sebacic acids [56]. The resulting copolymer was injected into the SEC apparatus, and the fraction eluting at 27 mL was collected. The fraction will be referred to as BABS27. Figure 11.10 reports the MALDI-TOF spectrum of BABS27. The mass spectral peaks are due to ions of the type:

$$\mathrm{H_3CO} - [R4]_m - [R8]_n - \mathrm{OCH_3} \ldots Li^+, \tag{11.21}$$

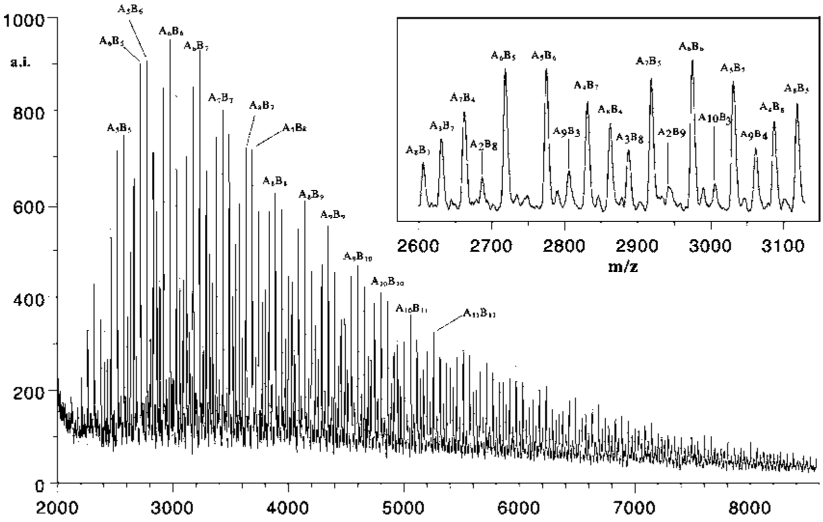


Fig. 11.10 MALDI-TOF spectrum of BABS27 copolymer. Reprinted from [56] with permission from the copyright holder Wiley 1998

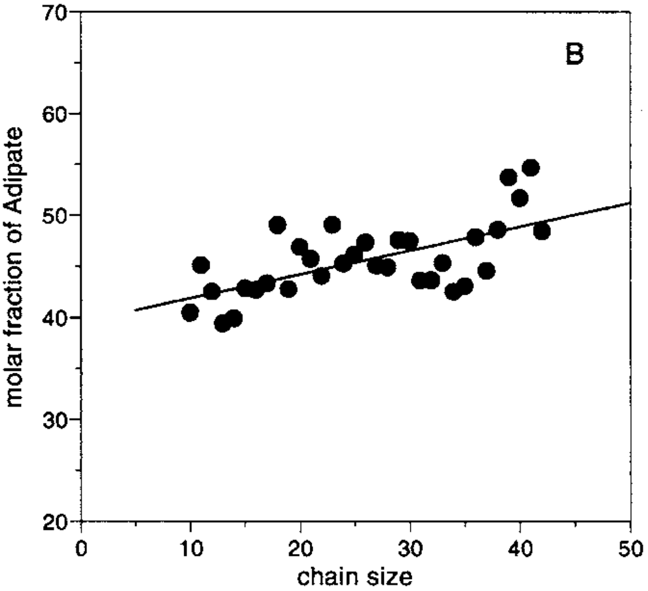


Fig. 11.11 Composition of the BABS27 copolymer as a function of chain size. Reprinted from [57] with the permission of the copyright holder Elsevier 2002

where $R4 = -\mathrm{CO(CH_2)_4COO(CH_2)_4O}-$, $R8 = -\mathrm{CO(CH_2)_8COO(CH_2)_4O}-$, and where m and n are the number of A and B units. The inset reports an expansion of a region of the spectrum. The peaks in the inset are due to chains with 11, 12, and 13 repeat units and the base peak in the spectrum is A_6B_6. MS peaks due to A_5B_6, A_6B_5, A_6B_7, and A_7B_7 are intense. Chain statistics was applied and the result was $c_\mathrm{A} = 0.46$ (c_A is the molar fraction of adipate units), which compares well with $c_\mathrm{A} = 0.47$ obtained from NMR analysis. Figure 11.11 reports the composition for the BABS27 copolymer as a function of chain size [57]. The points are the values determined experimentally. The line is the prediction of (11.11). The agreement between the two data sets is excellent.

References

1. P.C. Painter, M.M. Coleman, **Fundamentals of Polymer Science**, Technomic Publ., Lancaster, 1997
2. P. Rempp, E.W. Merril, **Polymer Synthesis, 2nd edition**, Huthig- Wepf, Basel, 1991
3. R.H. Boyd, P.J. Phillips, **The Science of Polymer Molecules**, Cambridge University Press, NY, 1993
4. S. Kuchanov, H. Slot, A. Stroeks, Prog. Polym. Sci. 29, 563–633 (2004)
5. E. Marechal, A. Fradet in edited by G. Allen. **Comprehensive Polymer Science, volume 5**, pp. 289–290, Pergamon Press, Oxford, 1989
6. M.S. Montaudo, C. Puglisi, F. Samperi, G. Montaudo, Macromolecules 31, 8666–8676 (1998)

7. M.S. Montaudo, G. Montaudo, Macromolecules 25, 4264–4280 (1992)
8. T.A. Speckhard, J.A. Miller, S.L. Cooper, Macromolecules 19, 1558–1567 (1986)
9. J.A. Miller, T.A. Speckhard, S.L. Cooper, Macromolecules 19, 1568–1574 (1986)
10. P.B. Zetterlund, R.G. Gosden, W. Weaver, A.F. Johnson, Polym. Int. 52, 749–756 (2003)
11. P.B. Zetterlund, W. Weaver, A.F. Johnson, Polym. React. Eng. 10, 41–57 (2002)
12. L.L. Jacobsen, W.H. Ray, J. Macromol. Sci. Rev. Macromol. Chem. Phys. C32, 407–519 (1992)
13. V.A. Vasnev, S.I. Kuchanov, In V.V. Korshak (editor), **Advances in Polymer Chemistry**, pp. 117–158, MIR Publishers, Moscow, 1986
14. M.S. Montaudo, J. Phys. Chem. B 108, 6288–6294 (2004)
15. G. Montaudo, M.S. Montaudo, In G. Montaudo, R.P. Lattimer (editors), **Mass Spectrometry of Polymers**, pp. 41–111, CRC Press, Boca Raton, 2003
16. D. Tillier, H. Lefebvre, M. Tessier, J.C. Blais, A. Fradet, Macromol. Chem. Phys. 205, 581–592 (2004)
17. G. Impallomeni, A. Steinbuchel, T. Lutke-Eversloh, T. Barbuzzi, A. Ballistreri, Biomacromolecules 8, 985–991 (2007)
18. G. Montaudo, M.S. Montaudo, F. Samperi, In G. Montaudo, R.P. Lattimer (editors), **Mass Spectrometry of Polymers**, pp. 419–521, CRC Press, Boca Raton, 2003
19. A.M. Kotliar, J. Polym. Sci. Macromol. Rev. 16, 367–395 (1981)
20. S. Fakirov (editor) **Transreactions in Condensation Polymers**, Wiley, Weinheim, 1999
21. A.D. Litmanovich, N.A. Plate, Y.V. Kudryavtsev, Progr. Polym. Sci. 27, 915-970 (2002)
22. N.A. Plate, A.D. Litmanovich, Y.V. Kudryavtsev, Polym. Sci. A 46, 1108–1140 (2004)
23. G. Montaudo, M.S. Montaudo, E. Scamporrino, D. Vitalini, Makromol. Chem. 194, 993–1001 (1993)
24. G. Montaudo, C. Puglisi, F. Samperi, Macromolecules 31, 650–661 (1998)
25. G. Montaudo, C. Puglisi, F. Samperi, Polym. Bull. 21, 483–488 (1989)
26. F. Samperi, C. Puglisi, R. Alicata, G. Montaudo, J. Polym. Sci. Part A Polym. Chem. 41, 2778–2793 (2003)
27. G. Montaudo, C. Puglisi, F. Samperi, F.P. Lamantia, J. Polym. Sci. Part A Polym. Chem. 34, 1283–1290 (1996)
28. C. Puglisi, F. Samperi, S. DiGiorgi, G. Montaudo, Macromolecules 36, 1098–1107 (2003)
29. Y.V. Kudryavtsev, E.N. Govorun, e-Polymers 033, 1–8 (2002)
30. Y.V. Kudryavtsev, E.N. Govorun, e-Polymers 063, 1–15 (2003)
31. M.S. Montaudo, Europ. Journ. Mass Spectrom. 14, 61–67 (2007)
32. L.H. Tung (editor), **Fractionation of Synthetic Polymers**, Marcel Dekker, NY, 1977
33. F. Francuskiewicz, **Polymer Fractionation**, Springer, Berlin, 1994
34. J. Stejskal, P. Kratochvil, Macromolecules 11, 1097–1103 (1978)
35. J. Podesva, J. Stejskal, P. Kratochvil, J. Appl. Polym. Sci. 49, 1265–1275 (1993)
36. K.L.L. Eersels, G. Groeninckx, Y. Mengerink, Sj. Van der Wal, Macromolecules 29, 6744–6749 (1996)
37. S. Weidner, J. Falkenhagen, R.P. Krueger, U. Just, Anal. Chem. 79, 4814–4819 (2007)
38. H. Lee, T. Chang, D. Lee, M.S. Shim, H. Ji, W.K. Nonidez, J.W. Mays, Anal. Chem. 73, 126–132 (2001)
39. M.S. Montaudo, Polymer 43, 6291–6298 (2004)
40. R. Yamadera, M. Murano, J. Polym. Sci. Part A1 5, 2259–2271 (1967)
41. J. Devaux, P. Godard, J.P. Mercier, R. Touillaux, J.M. Dereppe, J. Polym. Sci. Polym. Phys. Ed. 1982, 20 (1881)
42. R.W. Lenz, J.I. Jin, K.A. Feichtinger, Polymer 24, 327–334 (1983)
43. C.F. Ou, Europ. Polym. J. 38, 2405–2411 (2002)
44. I. Kreiser-Saunders, H.R. Kricheldorf, Macromol. Chem. Phys. 199, 1081–1087 (1998)
45. J. Kasperczyk, Macromol. Chem. Phys. 200, 903–910 (1999)
46. A. Muhlebach, R.D. Johnson, J. Lyerla, J. Economy, Macromolecules 21, 3117–3119 (1988)
47. M. Tessier, A. Fradet, e-Polymers 30, 1–7 (2003)
48. G. Montaudo, M.S. Montaudo, C. Puglisi, F. Samperi, Macromol. Chem. Phys. 196, 499–511 (1995)
49. C.C. Su, C.K. Shih, Colloid Polym. Sci. 283, 1278-1288 (2005)

50. H.R. Kricheldorf, W.E. Hull, J. Macromol. Sci. Chem. A11, 2281–2292 (1977)
51. M.S. Montaudo, Rapid Comm. Mass Spectrom. 13, 639–649 (1999)
52. A. Ballistreri, G. Montaudo, D. Garozzo, M. Giuffrida, M.S. Montaudo, Macromolecules 24, 1231–1236 (1991)
53. G. Montaudo, F. Samperi, M.S. Montaudo, Prog. Polym. Sci. 31, 277–357 (2006)
54. G. Montaudo, M.S. Montaudo, E. Scamporrino, D. Vitalini, Macromolecules 25, 5099–5107 (1992)
55. S. Carroccio, P. Rizzarelli, C. Puglisi, Rapid Commun. Mass Spectrom. 14, 1513–1522 (2000)
56. M.S. Montaudo, C. Puglisi, F. Samperi, G. Montaudo, Rapid Commun. Mass Spectrom. 12, 519–528 (1998)
57. M.S. Montaudo, Polymer 43, 1587–1597 (2002)

Index